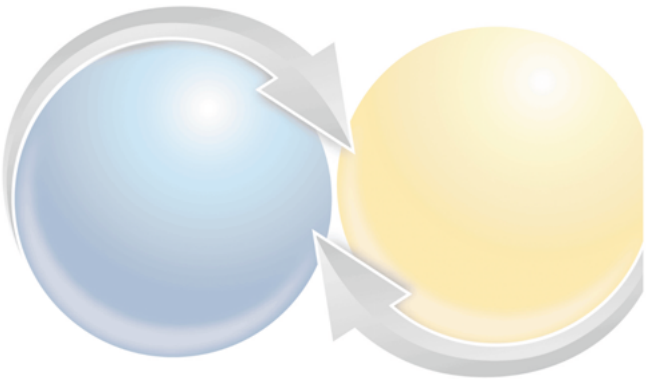
Livelink WCM Server

Installation Manual

This manual describes the installation of Livelink WCM Server. It provides information on the following topics:

- configuring the database
- configuring an LDAP directory service
- installing and deinstalling the WCM system and individual WCM components
- configuring the web server
- upgrading the WCM system



OPEN TEXT CORPORATION

Great Minds Working Together™

Copyright 2006 by Open Text Corporation.

The copyright to these materials and any accompanying software is owned, without reservation, by Open Text Corporation. These materials and any accompanying software may not be copied in whole or part without the express, written permission of Open Text Corporation.

Open Text Corporation is the owner of the trademarks Open Text, 'Great Minds Working Together', Livelink, and MeetingZone among others. This list is not exhaustive. All other products or company names are used for identification purposes only, and are trademarks of their respective owners. All rights reserved.

Open Text Corporation provides certain warranties and limitations in connection with the software that this document describes. For information about these warranties and limitations, refer to the license agreement entered into between the licensee and Open Text Corporation.

Contacting Us

Open Text Corporation
Corporate Headquarters
275 Frank Tompa Drive,
Waterloo, Ontario, Canada
N2L 0A1

(519) 888-7111

If you subscribe to our Customer Assistance Program or would like more information about the support program, write to Open Text Corporation's Customer Support at support@opentext.com or telephone (800) 540-7292 or (519) 888-9933. Our support hours are Monday through Friday, 8:30 a.m. to 8 p.m. (EST).

If you have comments or suggestions regarding this documentation, write to the Open Text Corporation Publications Group at documentation@opentext.com.

For more information about Open Text Corporation's products and services, visit our home page at http://www.opentext.com.

© 2006 IXOS SOFTWARE AG

Werner-v.-Siemens-Ring 20
85630 Grasbrunn, Germany

Tel.: +49 (89) 4629-0
Fax: +49 (89) 4629-1199
eMail: office@ixos.de
Internet: http://www.ixos.com

All rights reserved, including those regarding reproduction, copying or other use or communication of the contents of this document or parts thereof. No part of this publication may be reproduced, transmitted to third parties, processed using electronic retrieval systems, copied, distributed or used for public demonstration in any form without the written consent of IXOS SOFTWARE AG. We reserve the right to update or modify the contents. Any and all information that appears within illustrations of screenshots is

provided coincidentally to better demonstrate the functioning of the software. IXOS SOFTWARE AG hereby declares that this information reflects no statistics of nor has any validity for any existing company. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/) and software developed by the Apache Software Foundation (http://www.apache.org/).

Trademarks IXOS: IXOS SOFTWARE AG.

SAP®, R/3® and SAP ArchiveLink® are registered trademarks of SAP AG.

Microsoft®, Microsoft Windows NT® and the names of further Microsoft products are registered trademarks of Microsoft Corporation.

Acrobat Reader Copyright © 1987 Adobe Systems Incorporated. All rights reserved. Adobe and Acrobat are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.

Siebel® is a registered trademark by Siebel Systems, Inc.

Other product names are used only to identify the products and they may be registered trademarks of the relevant manufacturers.

Copyright © 2006 Gauss Interprise AG Hamburg, Gauss Interprise, Inc. Irvine, California. All rights reserved worldwide.

This document and the related software are property of Gauss Interprise AG or its suppliers and protected by copyright and other laws. They are distributed under licenses restricting their use, copying, distribution, and decompilation. Neither receipt nor possession of this document confers or transfers any right to reproduce or disclose any part of the contents hereof. No part of this document may be reproduced in any form by any means without prior written authorization of Gauss Interprise AG or Gauss Interprise, Inc.

Moreover, the regulations of the software license agreement apply to this documentation.

All brand names and trademarks mentioned are the property of their respective owners.

http://www.opentext.com/bridging/gauss.html

Program version: Livelink Web Content Management Server™ (Content Server) 9.5.1

Document version: En-01

Publication date: May 2006

Table of Contents

List of Figures 6

List of Tables 8

Chapter 1 Introduction 11

Installation Procedure 11

Installation Requirements 13

About this Document 16

Typographic Conventions 18

Chapter 2 Configuring the RDBMS 21

Configuring Oracle 22

Configuring MS SQL Server 2000 29

Chapter 3 Configuring the LDAP Directory Service 33

Concept of Collective Groups and Collective Roles 36

Creating Additional Object Classes and Attributes 38

Creating the LDAP Binding Profile and the WCM Administrator 53

Notes on Using Multiple LDAP Servers 55

Using SSL Connections to the LDAP Server 61

Chapter 4 Installing Livelink WCM Server 67

Installing a WCM System (Minimum System) 68

User-Defined Installation of the WCM System 106

Adding New Servers 110

Installing the Admin Client 113

Updating the License 115
Directory Structure after the Installation 117
Deinstalling the WCM System 121
Installation and Deinstallation via the Console 126
Starting and Stopping Servers 151
Log Files and Error Files 162

Chapter 5 Configuring the Web Server 165
Scenario 1 – Apache 2 with Tomcat 168
Scenario 2 – BEA WebLogic 8.1 170
Scenario 3 – MS Internet Information Server with Resin 172
The Precompile Script 188
Configuring Secure Access 189

Chapter 6 Upgrading Livelink WCM Server 213
Upgrade via the Graphical User Interface 213
Upgrade via Console 221
Steps Required after the Upgrade 225

Appendix A Product-Specific Information for LDAP Directory Services 229
Microsoft Active Directory 230
Novell eDirectory (NDS) 236
Sun ONE Directory Server 241
OpenLDAP 245

Glossary 251

Index 261

List of Figures

Fig. 1 –	Settings of the database server (SQL Server)	31
Fig. 2 –	Using several LDAP servers	57
Fig. 3 –	Using one LDAP server for several WCM systems	59
Fig. 4 –	Overview of installation options	67
Fig. 5 –	Installation options	71
Fig. 6 –	Specifying the license management options	73
Fig. 7 –	Parameters of the Administration server	74
Fig. 8 –	Parameters of the database to be used	77
Fig. 9 –	Parameters of the Livelink system to be used	81
Fig. 10 –	Parameters of the LDAP directory service to be used	83
Fig. 11 –	WCM-specific LDAP parameters	86
Fig. 12 –	Setting up the WCM administrator	89
Fig. 13 –	Installation variants	90
Fig. 14 –	Dialog box for configuring the servers	91
Fig. 15 –	Generating a web application for a Content server	95
Fig. 16 –	Installing a Content Miner Search server	99
Fig. 17 –	Installing a Lucene Search server	101
Fig. 18 –	Installing a Livelink Search server	103
Fig. 19 –	Selecting the components to be removed	124
Fig. 20 –	URL mapping in IIS	174
Fig. 21 –	IIS configuration for base authentication with anonymous access	181
Fig. 22 –	Configuring the VIPSecure.dll as a ISAPI filter (IIS)	183
Fig. 23 –	IIS configuration for base authentication with name and password	185
Fig. 24 –	IIS configuration for NTLM authentication	187
Fig. 25 –	Upgrading the version	216

Fig. 26 – Upgrading the Data Storage 219
Fig. 27 – Data storages selected for upgrade 220

List of Tables

Table 1 –	Values for Oracle database instances	23
Table 2 –	OIDs of the LDAP object classes for Livelink WCM Server	41
Table 3 –	OIDs of the LDAP attributes for Livelink WCM Server	41
Table 4 –	Entries in the `<mappings>` section	43
Table 5 –	Tags for controlling readability and writability of LDAP attributes	49
Table 6 –	Inverse LDAP attributes for the class `vipUser`	52
Table 7 –	Sample configuration for separate LDAP object classes and attributes	60
Table 8 –	Parameters for the keytool	63
Table 9 –	Entries in the `<license>` section	130
Table 10 –	Entries in the `<common>` section	131
Table 11 –	Entries in the `<admin>` section	132
Table 12 –	Entries in the `<server>` section	135
Table 13 –	Entries in the `<contentminer_server>` section	139
Table 14 –	Entries in the `<lucene_server>` section	140
Table 15 –	Entries in the `<livelink_search-server>` section	141
Table 16 –	Entries in the `<rdbms>` section	143
Table 17 –	Entries in the `<livelink> section`	144
Table 18 –	Entries in the `<ldap>` section	146
Table 19 –	Parameters in the server start scripts	154
Table 20 –	Parameters in the server stop scripts	157
Table 21 –	Log levels of Secure Access	201
Table 22 –	Entries in the `<update_vip>` section	222
Table 23 –	Entries in the `<update_rdbms>` section	224
Table 24 –	WCM attributes for the class `vip` (Active Directory)	232
Table 25 –	WCM attributes for the class `vip` (Active Directory)	233

Table 26 – WCM attributes for the classes `vipGroup` and `vipRole` (Active Directory) 234
Table 27 – WCM attributes for the class `vip` (NDS) 238
Table 28 – WCM attributes for the class `vipUser` (NDS) 239
Table 29 – WCM attributes for the classes `vipGroup` and `vipRole` (NDS) 240
Table 30 – WCM attributes for the class `vip` (Sun ONE) 242
Table 31 – WCM attributes for the class `vipUser` (Sun ONE) 243
Table 32 – WCM attributes for the classes `vipGroup` and `vipRole` (Sun ONE) 244
Table 33 – WCM attributes for the class `vip` (OpenLDAP) 246
Table 34 – WCM attributes for the class `vipUser` (OpenLDAP) 247
Table 35 – WCM attributes for the classes `vipGroup` and `vipRole` (OpenLDAP) 248

CHAPTER 1
Introduction

This installation manual describes how to install a WCM system and deals with the different installation variants and other aspects that must be taken into account during installation.

Installation Procedure

Setting up a WCM system does not only include the installation of Livelink Web Content Management Server™ (Livelink WCM Server for short). Additional components, such as a relational database management system, an HTTP server, and a JSP engine, are required for operating a content management system.

Before the installation of Livelink WCM Server

Depending on the IT infrastructure of your company, other software components must be installed or configured for the use with Livelink WCM Server before Livelink WCM Server can be installed.

For saving website data and (optionally) user data, for example, a relational database is required. You must set up the database before you install Livelink WCM Server. Alternately, you can use an existing LDAP directory service or a Livelink user administration for saving the user data.

- For information on installing and configuring a RDBMS, refer to chapter 2 “Configuring the RDBMS”.
- For information on integrating an LDAP directory service, refer to chapter 3 “Configuring the LDAP Directory Service”.
- For information on integrating a Livelink user administration, refer to the Livelink Integration Manual.

Installing Livelink WCM Server

The installation is carried out via a separate program, which can be used for different tasks:

- installing a new WCM system (see section “Installing a WCM System (Minimum System)” starting on page 68 and section “User-Defined Installation of the WCM System” starting on page 106)

> **Note:** Before installing the WCM system, read chapter “Concepts” of the Livelink WCM Server Administrator Manual to get an overview of the architecture of WCM systems.

- adding servers to an existing WCM system (see section “Adding New Servers” on page 110)
- installing the Admin client (see section “Installing the Admin Client” on page 113)
- updating the license for Livelink WCM Server (see section “Updating the License” on page 115)
- deinstalling the entire WCM system or individual components (see section “Deinstalling the WCM System” on page 121)

Livelink WCM Server can also be installed in command line mode, see chapter 4 “Installation and Deinstallation via the Console”.

For information on the installation logs, refer to section “Log Files and Error Files” on page 162.

After the installation of Livelink WCM Server

To ensure that all components of the WCM system function smoothly, other components, such as HTTP server and JSP engine, must be configured after the installation of Livelink WCM Server. For more information, refer to chapter 5 “Configuring the Web Server”.

For information on starting and stopping the WCM servers, refer to section “Starting and Stopping Servers” on page 151.

Upgrading Livelink WCM Server

You can use the installation program to upgrade an existing WCM system to a new program version. Upgrading comprises two steps: First, the program version is upgraded. The second step is upgrading the data storage.

For information on performing an upgrade, refer to chapter 6 “Upgrading Livelink WCM Server”.

Installation Requirements

Operating Livelink WCM Server requires the following components, which must be available before installation.

> **Note:** For information on the supported versions of the software components listed, refer to the Release Notes.

Java 2 Software Development Kit

The Java 2 SDK (also called JDK) is required for executing the WCM server programs. Since all WCM programs – including the installation – are implemented entirely in Java, the Java 2 SDK must be available before installing the WCM system.

HTTP server

To enable access to the generated pages in the Edit, QA, and Production views, an HTTP server (e.g. iPlanet Web Server, Apache HTTP Server, Microsoft Internet Information Server) must be installed on the computers hosting the relevant WCM servers.

The HTTP server must be configured in such a way that your WCM system can make the managed information available. For information on how to configure the HTTP server, please refer to the documentation supplied by the manufacturer and to chapter 5 “Configuring the Web Server”.

JSP engine

For using the following components, you require a JSP engine that must be registered with the HTTP server: Content client, Content client (Classic), dynamic deployment, Search servers, Secure Access, InSite Editing, and Portal Manager API.

Information on how to configure the JSP engine can be found in the manufacturer’s documentation. For information on the WCM-specific configuration of the JSP engine, refer to chapter 5 “Configuring the Web Server”.

> **Note:** If you use the Portal Manager API under UNIX, please make sure that the number of file descriptors available to the JSP process amounts to at least 1024.

Mail server

The WCM system uses e-mails to notify responsible persons. Therefore a mail server must be installed and configured. Livelink WCM Server uses the SMTP protocol to send e-mails; this protocol is supported by most mail servers or can be activated by means of an appropriate gateway.

Relational database management system

For saving the WCM objects and (optionally) the user data, you require an external, relational database management system (RDBMS).
Livelink WCM Server supports the RDBMS Oracle and MS SQL Server 2000. For detailed information on configuring the RDBMS, refer to chapter 2 “Configuring the RDBMS”.

LDAP directory service (directory server)

By using an LDAP directory service (LDAP = Lightweight Directory Access Protocol), such as Sun ONE Directory Server, Novell eDirectory, and Microsoft Active Directory, it is possible to integrate existing user administration facilities.

Chapter 3 “Configuring the LDAP Directory Service” provides detailed information on this topic.

Optional components

Various components can be added to the WCM system:

- Firewalls

  The WCM system may be distributed over two or more physical computers. Several proxy Content servers and a proxy Admin server can be set up outside a firewall (outside the corporate network). In this case, the appropriate HTTP or VIPP ports of the servers must be enabled.

  An example of how to install such a system is described in section “User-Defined Installation of the WCM System” starting on page 106.

- Server-side applications

  In connection with Livelink WCM Server, you can use all facilities offered by web technology, e.g. Java applets, CGI scripts, server-side includes (SSI), and servlets. Some of these require installation and/or configuration of the HTTP server and/or JSP engine. These do not concern the WCM system and therefore do not require any changes to the WCM system.

About this Document

This manual describes the installation of Livelink WCM Server as well as the steps to be performed before and after the installation. The installation should be performed by experienced system administrators only. For installing a WCM system, you should have a sound knowledge of the following fields:

- installation and configuration of a web server (HTTP server and JSP engine)
- standard methods of system administration, e.g. configuring and editing user administration systems, directory trees, and files
- administration of relational database management systems
- administration of LDAP directory servers (optional)

In addition to this Installation Manual, the following sources provide information:

- Livelink WCM Server Administrator Manual: This document describes the configuration and administration of a WCM system from the point of view of an administrator.
- Content Client User Manual: This manual provides detailed instructions on all tasks involved in editing websites in the WCM system.

- WCM Java API Programmer's Manual: This document contains information on interfaces, classes, and methods of the Java programming interface (WCM Java API), which enables you to use the functionality of the WCM servers via external programs.
- Portal Manager API Programmer's Manual: This manual contains detailed information on developing dynamic and personalized websites on the basis of the Portal Manager API.
- WCM WebServices Programmer's Manual: This manual contains information on using the functions of Livelink WCM Server via Web Services.
- Online help: For the clients of Livelink WCM Server (Content client, Admin client, and Content Workflow Modeler), online helps are available, which can be called at any time.

The contents of this manual are organized as follows:

- Chapter 2 “Configuring the RDBMS” contains information on configuring the relational database management system used together with Livelink WCM Server.
- Chapter 3 “Configuring the LDAP Directory Service” describes the configuration of an LDAP directory service.
- Chapter 4 “Installing Livelink WCM Server” explains the procedure of installing a new WCM system and the other functions of the WCM installation program.
- Chapter 5 “Configuring the Web Server” contains sample configurations of HTTP servers and JSP engines for the use with Livelink WCM Server.
- Chapter 6 “Upgrading Livelink WCM Server” describes how to upgrade the system using the WCM installation program.
- Appendix A contains product-specific information on integrating different LDAP directory services.

Typographic Conventions

The following conventions are used in the text to draw attention to program elements, etc.:

Item	Font or Symbol	Examples
Program interface such as menu commands, windows, dialog boxes, field and button names	*Menu → Entry*	*File → Create*
Paths to directories, file and directory names	**Drive:\Directory\File name**	**D:\WCM\ admin.bat**

Quotations from program code or configuration files	`Code quotations`	`<head>`
		`<title>heading </title>`
		`</head>`

Variables, i.e. placeholders for specific elements	{variable}	{WCM installation directory}

Important **information** and **warnings** are enclosed in gray boxes. Make sure to read such information to avoid losing data or making errors when using and managing WCM systems.

CHAPTER 2
Configuring the RDBMS

For operating Livelink WCM Server, a relational database management system is required. The WCM system saves the website data and (optionally) the user data in the database. Livelink WCM Server supports the RDBMS Oracle and MS SQL Server 2000.

For using Livelink WCM Server, you can install a new RDBMS or configure an existing RDBMS accordingly. This must be done **before** the installation of Livelink WCM Server.

This chapter contains information on the WCM-specific configuration of the RDBMS:

- "Configuring Oracle" starting on page 22
- "Configuring MS SQL Server 2000" starting on page 29

Notes

- For information on the RDBMS versions supported, refer to the Release Notes of Livelink WCM Server.
- The RDBMS is installed by means of the installation program supplied by the respective manufacturer. For information on the installation procedure, refer to the manufacturer's documentation.
- The configuration of the RDBMS strongly depends on the IT infrastructure of your company. Thus, no generally valid information can be provided here. We recommend you to develop the database architecture in cooperation with the Professional Services Group of Gauss Interprise AG.

Configuring Oracle

Notes

- For performance reasons, the RDBMS and the WCM servers should not be installed on the same computer.
- During the operation of the WCM system, the table contents may change considerably. For this reason, you should execute the function *Compute Statistics* on the WCM tables at regular intervals. How frequently you perform this function depends on the frequency of changes to the WCM tables. For a start, we recommend that you execute the function once a month. You should compute the statistics when the performance of the WCM system declines.

  The statistics can be computed conveniently by means of the Oracle administration program “Enterprise Manager Console”. After installing Oracle in Windows, this program can be started via *Start* → *Programs* → *Oracle* → *Enterprise Manager Console*. After logging in to the database, choose *Tools* → *Database Tools* → *Analyze* → *Compute Statistics*.
- Make sure that the version of the database driver used corresponds to the version of the database.

To use an existing RDBMS in connection with Livelink WCM Server, some settings must be modified in the RDBMS. These include:

- database instances (see the following section)
- use of open cursors (see section “Use of Open Cursors” on page 24)
- maximum number of parallel processes (see section “Parallel Server Processes” on page 26)

Moreover, you must create a tablespace and a user for Livelink WCM Server (see section “Creating the Database User and Tablespace in Oracle” starting on page 26).

Configuring the Database Instance(s)

When configuring the database instance(s), you must modify certain configuration parameters. Enter the following values:

Table 1 – Values for Oracle database instances

Parameter	Value	Changeable
Memory tab → *Shared Pool*	150 MB	✓
Character Sets tab → *Database Character Set*	UTF-8	
Character Sets tab → *National Character Set*	UTF-8	
DB Sizing tab → *Block Size*	8 KB	
Archive tab → *Archive Log Mode*	activate	✓
All Initialisation Parameters button → *log_checkpoint_interval*	100000	✓
All Initialisation Parameters button → *open_cursors*	1000	✓
All Initialisation Parameters button → *parallel_max_servers*	5	✓
All Initialisation Parameters button → *processes*	500	✓

For the other parameters, you can adopt the default settings.

Notes

- The parameters “Database Character Set” and “National Character Set” **cannot** be changed subsequently! By entering “UTF-8”, you set the database to Unicode. Alternately, you can specify “ISO 8859”. In this case, however, Unicode cannot be used in the metadata of the WCM objects, i.e. Asian languages, for example, will not be supported. You should configure the database for UTF-8, even if Latin-1 encoding is used in your website. The additional memory space required for UTF-8 encoding is relatively small. This way, you do not have to migrate the database later.
- Please note the general information on using Unicode with Livelink WCM Server in chapter “Concepts” of the Livelink WCM Server Administrator Manual.
- We recommend that you set the database to the archive log mode. This mode makes it easier to restore the database in the case of errors. However, performance may be slightly reduced in this mode.

Use of Open Cursors

A database cursor is an area in the database memory for temporarily storing internal information. Livelink WCM Server also opens cursors in the database. These are required for two purposes in particular:

- The database assigns a cursor to each Oracle statement which processes more than one line (both read and write access). By means of these cursors, results can be processed line by line. After Livelink WCM Server has evaluated the request, the cursor is closed, i.e. the reserved memory is released.
- Each statement that Livelink WCM Server sends to the database (in the form of prepared statements) is assigned a cursor. In this case, the cursor speeds up the execution of the statement.

Livelink WCM Server uses a high number of prepared statements and thus causes many opens cursors. The connections from Livelink WCM Server to the database are managed in JDBC pools. In the Admin client, you can determine the maximum number of open connections in the settings of the JDBC pools. Each connection can open the maximum number of cursors specified in the *Extended settings* of the pool. If the pool is used by several servers, each server can use the maximum number of connections specified in the pool. This results in a very high, theoretically possible number of open cursors required for Livelink WCM Server. However, this value is never reached in practice.

Modify the value for the maximum number of open cursors per session in the database according to the configuration of your WCM system (number of servers and number of connections per JDBC pool). This value is only limited by the given operating system resources. For a start, set the parameter “open_cursors” to 1000.

Notes

- We recommend that you specify a high value for the maximum number of simultaneously open cursors.
- If the specified number of open cursors is exceeded by Livelink WCM Server, Oracle generates an error message. The affected action in the WCM system fails.
- The computer on which the database is installed must have a sufficient performance.
- In the Admin client, you can enter the maximum number of open cursors per JDBC pool. The values set here have already been optimized for Livelink WCM Server and should only be changed in cooperation with Gauss Interprise AG.

  If Oracle or Livelink WCM Server repeatedly display error messages about an insufficient number of cursors, please contact the Technical Support of Gauss Interprise AG.

Parallel Server Processes

The initialization parameter “parallel_max_servers” specifies the maximum number of parallel execution processes and parallel recovery processes for a database instance. Set the parameter value to 5.

As demand increases, Oracle increases the number of processes from the number created at instance startup up to this value.

Depending on the performance of the computer, each server instance can have a certain number of processes. This number is determined via the parameter “processes”. This parameter should also be modified (value = 500).

Creating the Database User and Tablespace in Oracle

In the database, the data of the WCM system is managed in a so-called tablespace. After the configuration of the RDBMS, you must create a tablespace and a database user for the WCM system.

Notes:

If several WCM servers save their data in the same database and, in particular, in the same tablespace, this can result in bottlenecks in the system resources of the database machine. For this reason, each WCM system should access a database of its own. If there are two or more installations of the WCM system, a separate tablespace and a separate database user must be created for each installation.

The proxy Content servers of the WCM system can also be configured to have a separate data storage. In this way, the workload on the database of the master Content server is decreased.

You can use the Enterprise Manager Console or SQL commands to create the tablespace and database user. First create the tablespace and then the database user.

Tablespace size

The tablespace for the WCM data should have a size of at least 500 MB. You have a free choice of name. Moreover, a temporary tablespace is required, which must have a size of at least 70 MB. The required size of the tablespace can vary strongly. It is influenced by the following factors:

- size of the content managed with Livelink WCM Server
- amount of links between the objects
- frequency of changes to objects and the resulting number of object versions
- number of archived versions (can be controlled via the utility “Database clean-up”, see Livelink WCM Server Administrator Manual)

Privileges and assignments of the user

The database user for the WCM system should have the roles “CONNECT” and “RESOURCE”.

If you do not want to use the roles mentioned above, assign the user the following system privileges:

- CREATE TABLE
- CREATE TRIGGER
- CREATE PROCEDURE
- CREATE SESSION

In addition to these privileges, the user requires the UNLIMITED TABLESPACE system privilege or the according quota on the tablespace.

Assign the database user the WCM tablespace as “Default Tablespace” and the temporary tablespace as “Temporary Tablespace”.

Configuring tablespace and user

You can use the Enterprise Manager Console to set up the table space and the database user. For further information, refer to the online help of the Enterprise Manager Console.

To configure the tablespace and the database user by SQL commands:

1. Start the database front end SQL Plus. The start file for Windows **sqlplusw.exe** is located in the directory **{Oracle installation directory}\bin**.
2. Log in with the user ID of the system administrator.

   Enter the appropriate password. In the field *Host String*, the following entries are possible:

   - The field remains empty: The system looks for a database on the local computer.
   - name of the database (e.g. “wcmdb”): The system looks for a database with this name on the local computer.
   - {database name}_{name of the database host}.{domain}, e.g. wcmdb_dbserver.company.example: The system looks for a database with this name on the specified computer.

The SQL commands for creating the tablespace and the database user might look like this:

```
SQL> CREATE TABLESPACE wcmspace
2    DATAFILE '{Oracle installation directory}\oradata\
                 {database name}\wcmspace.ora'
3    SIZE 500M
4    REUSE
5    AUTOEXTEND ON;

Tablespace created.

SQL> CREATE USER wcmuser
2    IDENTIFIED BY wcm123
3    DEFAULT TABLESPACE wcmspace
```

```
4    TEMPORARY TABLESPACE temp;

User created.

SQL> GRANT connect TO wcmuser;

Grant succeeded.

SQL> GRANT resource TO wcmuser;

Grant succeeded.

SQL>
```

Configuring MS SQL Server 2000

When installing Livelink WCM Server, you must specify a JDBC driver for MS SQL Server 2000 (see section “Setting RDBMS Parameters” on page 76). For information on the JDBC drivers supported, refer to the Release Notes of Livelink WCM Server.

The following section describes the changes that are required for an existing MS SQL Server 2000 installation.

> **Note:** Due to the restrictive lock mechanism of MS SQL Server, we recommend that you set up separate databases for the master Content server and the proxy Content servers. This ensures that read access of the proxy Content servers is not blocked by actions that are performed on the master Content server.

Creating a database and a user for Livelink WCM Server

We recommend that you create a separate database for the data of the WCM system.

Also, create a separate user for Livelink WCM Server. This user must be assigned to the database created for Livelink WCM Server and belong to the following groups and roles:

- Server Role: Public
- Group: db_ddladmin

The user must be able to log in via the SQL server authentication, i.e. the user must have been created by means of the database user administration. The JDBC drivers for MS SQL Server do not support Windows authentication.

During the installation of Livelink WCM Server, this user is entered as the owner of the database, see section “Setting RDBMS Parameters” on page 76.

Properties of the database

In the database, *Authentication SQLServer & Windows* must be set as authentication method.

Moreover, the database must allow nested triggers. In the properties of the database server, select the check box *Allow triggers to be fired which fire other triggers (nested triggers)* on the *Server Settings* tab.

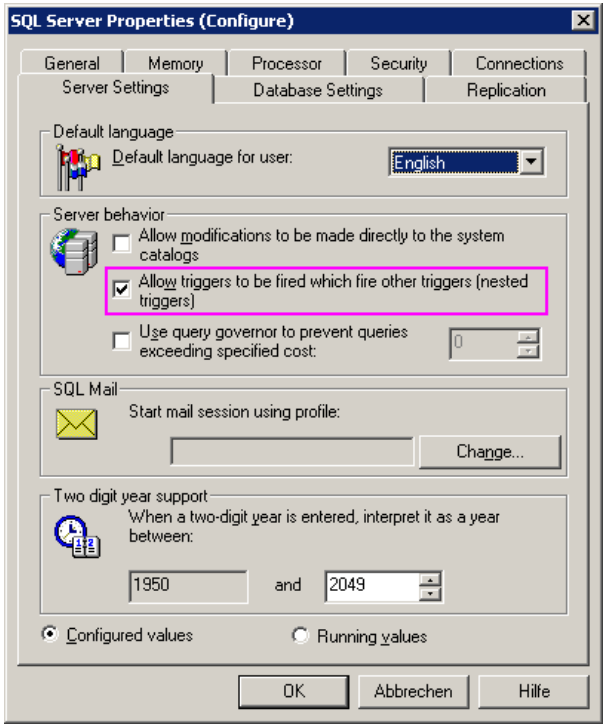


Fig. 1 – Settings of the database server (SQL Server)

CHAPTER 3

Configuring the LDAP Directory Service

As an alternative to an RDBMS or Livelink, data of users, groups, and roles can be saved in an LDAP directory service. This chapter describes the preparations required for using an LDAP directory service in connection with Livelink WCM Server.

Livelink WCM Server can read the following WCM information from the directory service and store it there:

- users with name, user ID, password, e-mail address, language, and substitute
- groups with name, e-mail address, and assigned users (static groups)
- roles with name, e-mail address, and assigned users (static roles)
- websites and functional areas assigned to the users, groups, and roles
- default object rights of users, groups, and roles

Notes on LDAP access

- The configuration of the LDAP directory service highly depends on the IT infrastructure of your company. For this reason, this chapter does not provide detailed instructions for setting up an LDAP directory service. Only the basic configuration will be described. We recommend that you develop the LDAP architecture in cooperation with the Professional Services Group of Gauss Interprise AG.
- If the LDAP directory service manages many users with numerous attributes, of which only some users are to access the WCM system and for which only some attributes are relevant for the WCMS, we recommend that you offload these users with the attributes relevant for the WCMS to a separate directory (e.g. by means of the replication mechanism of the LDAP directory service). This makes access to the WCM system faster (see section “Speeding up LDAP Requests” on page 51).
- The user identification of WCM users may only contain the following characters: a–z, A–Z, 0-9 and - (hyphen), _ (underscore), . (dot), \ (backslash), and & (ampersand).
- WCM users must have unique IDs. The names of groups and roles must also be unique. In the WCM system, user IDs as well as group and role names are case sensitive.
- The hierarchical structure of the LDAP directory service is not represented in the Admin client of the WCM system. The users are shown in a flat list. However, the LDAP position is displayed as a property of the user in the Admin client.

Supported LDAP servers

For information on the LDAP servers supported, refer to the Release Notes of Livelink WCM Server.

Configuring the LDAP directory service for Livelink WCM Server

To enable access to the LDAP directory service for Livelink WCM Server, the following preparations are required:

1. Create the WCM object classes and attributes in LDAP (see section “Creating Additional Object Classes and Attributes” starting on page 38)
2. Configure the LDAP connection for Livelink WCM Server and create the WCM administrator in the LDAP directory service (see section “Creating the LDAP Binding Profile and the WCM Administrator” starting on page 53).

Using multiple LDAP servers

If you want to allow access to the WCM system for users, groups, or roles which are managed in different LDAP servers, please read the information in section “Notes on Using Multiple LDAP Servers” on page 55.

There you will also find information on using a common LDAP directory service for several WCM systems.

Using SSL for the connections to the LDAP server

Secure connections (SSL = Secure Socket Layer) can be used between the WCM system and the LDAP server. For further information, refer to section “Using SSL Connections to the LDAP Server” on page 61.

Concept of Collective Groups and Collective Roles

This section introduces you to an extended concept of LDAP groups and roles. This concept is proprietary and not part of the LDAP standard. Livelink WCM Server differentiates between two types of groups and roles:

- standard groups and roles
- collective groups and roles

Standard groups and roles

Users are assigned to the standard groups and roles by means of the LDAP attribute `member`.

Collective groups and roles

Collective groups and roles are based on the organizational units represented in the LDAP directory service. Collective groups/roles may contain all types of WCM principals (users, groups, roles).

The following users are assigned to a collective group/role:

- All users assigned to the collective group/role via the LDAP attribute `member` (corresponds to the behavior for standard groups/roles). In this case, the assignment is explicit.
- All WCM users located below the collective group/role in the LDAP tree. In this case, the assignment is implicit.
- All users from standard groups/roles that are referenced by the LDAP attribute `member` and that are located below the collective group/role in the LDAP tree. In this case, the assignment is implicit.

Notes on collective groups/roles

- The users that are implicitly assigned to the collective group/role are visible in the Admin client. However, you cannot edit the **implicit** assignments via the Admin client or the WCM Java API. Use the administration tool of the LDAP directory service.
- In Livelink WCM Server, there is no visible differentiation between implicitly and explicitly assigned users. This information can only be retrieved from the LDAP directory service.
- You cannot use an alias to establish the relation “below the collective group/role in the LDAP tree”.

Collective groups/roles option

By means of the option *Collective groups/roles* in the settings of the LDAP pool, you can determine that implicit assignments of users to groups and roles are considered in the WCM system. This setting is available to you in the installation program (see section “Setting the Parameters for the LDAP Directory Service” on page 82) or in the Admin client.

This setting applies globally to the LDAP pool. If you set up several LDAP pools in your WCM system, make sure that this setting is identical for pools whose LDAP contexts overlap.

Creating Additional Object Classes and Attributes

Before installing the WCM system, you must create additional object classes and attributes in the LDAP server so that you can manage users, groups, and roles of the WCM system on the basis of the LDAP directory service. For this purpose, you need the base DN (search node), an LDAP binding profile including password, and the LDAP URL.

Notes

- The user data is **not** copied. The users are managed directly in the LDAP directory service. Thus, no synchronization is required.
- For information on how to create object classes and attributes in the LDAP directory service, refer to the documentation of the product you use.
- For detailed information on the LDAP directory services Microsoft Active Directoy, Novell eDirectory, Sun ONE Directory Server, and OpenLDAP, refer to appendix A “Product-Specific Information for LDAP Directory Services”.

For identifying the LDAP object classes and attributes, so-called OIDs (object identifiers) are used. You can use both custom OIDs according to the organization of your LDAP directory service or the OIDs of Gauss Interprise AG. For an overview of the OIDs, refer to section “LDAP OIDs of Gauss Interprise AG” on page 41.

Object classes for Livelink WCM Server

The following object classes are required for Livelink WCM Server:

- class for storing user information
  default name = `vipUser`
- class for storing group information
  default name = `vipGroup`

- class for storing role information
  default name = `vipRole`

These object classes should be derived from a common parent class `vip` whose parent class is `top`. You can also use custom names for the object classes. For the WCM user information to be saved correctly, these custom names must be specified when installing Livelink WCM Server and when creating pools for the connections to the LDAP directory service.

The following WCM attributes are required for the WCM object classes:

Attributes for the object class `vip`

- `vipAccess` = permission to access the WCM system
- `vipWebsite` = assigned websites
- `vipFuncarea` = functional areas
- `vipRights` = default object rights
- optional: `vipType` = principal type (user, group, or role)

Attributes for the object class `vipUser`

- `cn` = common name = user name
- `uid` = unique user ID = user ID for the WCM system
- `email` = the user's e-mail address
- `vipLanguage` = the user's language setting (locale)
- `vipUserPassword` = user password in LDAP and for the WCM system
- `initPassword` = indicates whether the user will be prompted to enter a new password the next time the user logs in
- `trustedLogin` = trusted login without password
- `vipSubstitute` = substitute, "distinguished name" of a WCM user

- `hclProfiles` = profile for the Content client
- `vipDomain` = the user's domain (when using Secure Access and the login method "ntlm")

Attributes for the object classes `vipGroup` and `vipRole`

- `cn` = common name = unique group or role name
- `member` = "distinguished names" of the assigned users
- `e-mail` = e-mail address of the group or role

The following attributes are by default used as naming attributes for the "distinguished names" of users, groups, and roles:

- `uid` of the class `vipUser` = unique user ID
- `cn` of the class `vipGroup` = unique group name
- `cn` of the class `vipRole` = unique role name

Notes

- The value of the naming attribute of a WCM user **must** be unique.
- Livelink WCM Server is not able to evaluate multi-valued "relative distinguished names" (RDN).
- The exact procedure for creating the object classes and attributes depends on the LDAP directory service used. Depending on the product, the WCM attributes must be mapped to LDAP attributes (see section "Mapping WCM Attributes to LDAP Attributes" on page 42). For this reason, note the information on the supported LDAP directory services in appendix A "Product-Specific Information for LDAP Directory Services".
- For working in the Admin client, the general rule applies that attributes that are part (name component) of the DN may not be changed.
- WCM attributes are case sensitive.

LDAP OIDs of Gauss Interprise AG

Each LDAP schema is identified by an OID (object identifier) that is unique worldwide. These OIDs are assigned by a central organization (IANA). Unique OIDs are required especially for using the Simple Network Management Protocol (SNMP).

For the object classes and attributes added in the LDAP directory service, you can use the OIDs of Gauss Interprise AG or register your own OIDs. The following tables contain the OIDs of Gauss Interprise AG.

The organizational OID of Gauss Interprise AG is 1.3.6.1.4.1.2027. To this number, 2.1 is added for the LDAP attributes and 2.2 is added for the LDAP object classes of Livelink WCM Server. The individual attributes and object classes are registered in a flat list.

Table 2 – OIDs of the LDAP object classes for Livelink WCM Server

Object class	OID
vip	1.3.6.1.4.1.2027.2.2.8.1
vipUser	1.3.6.1.4.1.2027.2.2.8.2
vipGroup	1.3.6.1.4.1.2027.2.2.8.3
vipRole	1.3.6.1.4.1.2027.2.2.8.4

Table 3 – OIDs of the LDAP attributes for Livelink WCM Server

Attribute	OID
vipType	1.3.6.1.4.1.2027.2.1.1
vipAccess	1.3.6.1.4.1.2027.2.1.2
vipRights	1.3.6.1.4.1.2027.2.1.3
vipFuncarea	1.3.6.1.4.1.2027.2.1.4

Attribute	OID
vipWebsite	1.3.6.1.4.1.2027.2.1.5
vipSubstitute	1.3.6.1.4.1.2027.2.1.6
initPassword	1.3.6.1.4.1.2027.2.1.7
trustedLogin	1.3.6.1.4.1.2027.2.1.8
hclprofiles	1.3.6.1.4.1.2027.2.1.10
vipUserpassword	1.3.6.1.4.1.2027.2.1.11
vipDomain	1.3.6.1.4.1.2027.2.1.12
vipSubstituteOf	1.3.6.1.4.1.2027.2.1.13
vipLanguage	1.3.6.1.4.1.2027.2.1.14
vipMemberOf	1.3.6.1.4.1.2027.2.1.15

Mapping WCM Attributes to LDAP Attributes

When setting up the object classes and attributes in the LDAP server, the following cases may occur:

- If some of the attributes listed above already exist in the LDAP server, they can simply be assigned to the WCM classes provided they have the right semantics and syntax.
- If there are attributes that already exist in the LDAP server and have the same name, but invalid values, you must create additional user-defined attributes.

- If some of the attributes that already exist in the LDAP server have valid values, but different names, map the required WCM attributes to the existing LDAP attributes. The following sections describe how to map the attributes.

Before the WCM system is installed, you can map the attributes in the file **defaults.xml**. This file is located in the **\installation** directory on the WCM CD and contains the necessary parameters for the installation.

After the installation, you can configure LDAP mappings in the **ldapmapping.xml** file, which is located in the **\config** directory of the WCM installation directory. This file contains a separate section for each LDAP pool.

The following table describes the section of the **defaults.xml** or **ldapmapping.xml** file that contains the mappings.

Table 4 – Entries in the <mappings> section

Entry in the XML file	Explanation
<CN> <vipattr>cn</vipattr> <ldapattr>cn</ldapattr> </CN>	User name or unique group or role name. **Note:** If you change this mapping, you have to modifiy the value of the <ldapattr> attribute in the mapppings <LDAP_USER_NAME>, <LDAP_GROUP_NAME>, and <LDAP_ROLE_NAME> accordingly.

Entry in the XML file	Explanation
<LDAP_USER_NAME> <vipattr>userclass.name</vipattr> <ldapattr>cn</ldapattr> <ldapread>false</ldapread> </LDAP_USER_NAME> <LDAP_GROUP_NAME> <vipattr>groupclass.name</vipattr> <ldapattr>cn</ldapattr> <ldapread>false</ldapread> </LDAP_GROUP_NAME> <LDAP_ROLE_NAME> <vipattr>roleclass.name</vipattr> <ldapattr>cn</ldapattr> <ldapread>false</ldapread> </LDAP_ROLE_NAME> <READ_PRINCIPAL_NAME> <vipattr>principal.name</vipattr> <ldapattr>sn</ldapattr> <ldapread>false</ldapread> <ldapwrite>false</ldapwrite> </READ_PRINCIPAL_NAME>	These mappings are required in order to use different attributes for the names of users, groups, and roles. For further information, see section “Using different attributes for the principal names” on page 49.
<INIT_PASSWORD> <vipattr>initPassword</vipattr> <ldapattr>initPassword</ldapattr> </INIT_PASSWORD>	Indicates whether the user will be prompted to enter a new password the next time the user logs in

Entry in the XML file	Explanation

`<LANGUAGE>`
`<vipattr>language</vipattr>`
`<ldapattr>language</ldapattr>`
`</LANGUAGE>`
	Language setting of the user

`<LDAP_OBJECTCLASS>`
`<vipattr>objectclass</vipattr>`
`<ldapattr>objectClass</ldapattr>`
`</LDAP_OBJECTCLASS>`
	Type of the LDAP entry

`<VIP_TYPE>`
`<vipattr>viptype</vipattr>`
`<ldapattr>vipType</ldapattr>`
`</VIP_TYPE>`
	Principal type (user, group, or role)

`<USER_ID>`
`<vipattr>userid</vipattr>`
`<ldapattr>uid</ldapattr>`
`</USER_ID>`
	Unique user ID for the WCM system

`<MAIL>`
`<vipattr>email</vipattr>`
`<ldapattr>mail</ldapattr>`
`</MAIL>`
	The principal's e-mail address

Entry in the XML file	Explanation
<TRUSTED_LOGIN> <vipattr>trustedLogin</vipattr> <ldapattr>trustedLogin</ldapattr> </TRUSTED_LOGIN>	Trusted login without password
<USER_PASSWORD> <vipattr>vipUserpassword</vipattr> <ldapattr>userPassword</ldapattr> </USER_PASSWORD>	User password in LDAP and for the WCM system
<VIP_ACCESS> <vipattr>vipAccess</vipattr> <ldapattr>vipAccess</ldapattr> </VIP_ACCESS>	Permission to access the WCM system
<VIP_FUNCAREAS> <vipattr>vipFuncarea</vipattr> <ldapattr>vipFuncarea</ldapattr> </VIP_FUNCAREAS>	Functional areas assigned to the principal
<VIP_MEMBERS> <vipattr>member</vipattr> <ldapattr>member</ldapattr> </VIP_MEMBERS>	“distinguished names” of the users assigned to the group or role

Entry in the XML file	Explanation

`<VIP_RIGHTS>`
`<vipattr>vipRights</vipattr>`
`<ldapattr>vipRights</ldapattr>`
`</VIP_RIGHTS>`
	Default object rights of the principal

`<VIP_SUBSTITUTE>`
`<vipattr>vipSubstitute</vipattr>`
`<ldapattr>vipSubstitute</ldapattr>`
`</VIP_SUBSTITUTE>`
	Substitute, “distinguished name” of a WCM user

`<VIP_WEBSITES>`
`<vipattr>vipWebsite</vipattr>`
`<ldapattr>vipWebsite</ldapattr>`
`</VIP_WEBSITES>`
	Websites assigned to the principal

`<HCL_PROFILES>`
`<vipattr>hclProfiles</vipattr>`
`<ldapattr>hclProfiles</ldapattr>`
`</HCL_PROFILES>`
	Profile of the user in the Content client

`<VIP_DOMAINS>`
`<vipattr>vipDomain</vipattr>`
`<ldapattr>vipDomain</ldapattr>`
`</VIP_DOMAINS>`
	The user’s domain (when using Secure Access and the login method “ntlm”)

Controlling readability and writability of the LDAP attributes

Each entry in the `<mappings>` section can be extended by the tags `<ldapread>` and `<ldapwrite>`. By means of these tags, you control whether the attribute values can be read from the LDAP server and saved in the LDAP server in case of changes. These tags can be used to transfer the read and/or write protection of LDAP attributes to the WCM system (e.g. for passwords).

Notes:

The Admin client does not show whether attributes are readable and/or writable. This information is only contained in the mapping file.

For attributes for which the tag `<ldapread>` has the value `false` (i.e. which are not read from the LDAP server), no default values are set in the WCM system.

Example

```
<MAIL>
      <vipattr>email</vipattr>
      <ldapattr>mail</ldapattr>
      <ldapread>true</ldapread>
      <ldapwrite>false</ldapwrite>
</MAIL>
```

The following table illustrates the possible combinations of the two tags.

Table 5 – Tags for controlling readability and writability of LDAP attributes

Value of <ldapread>	Value of <ldapwrite>	Description
true	true	Default setting (does not have to be set explicitly in the mapping file) The attribute values can be read by Livelink WCM Server and saved in the LDAP directory service in the case of changes.
true	false	The attribute values can be read by Livelink WCM Server. If they are changed in the WCM system, the changed values cannot be saved in the LDAP directory service. The attribute values can only be changed directly in the LDAP directory service.
false	true	The attribute values are not read by Livelink WCM Server. Changes can, however, be saved.
false	false	It is neither possible to read nor to save the attribute values.

Using different attributes for the principal names

By default, the CN attribute is used for the names of user and group/role entries. You may, however, use different attributes, e.g. the CN attribute for users and the SN attribute for groups/roles. In this case, the following mappings are required:

- <LDAP_USER_NAME>, <LDAP_GROUP_NAME>, <LDAP_ROLE_NAME>

  These mappings are used for searches and writes accesses. Read access must **always** be deactivated. The value of the <ldapattr> attribute must be set to the desired value for the group mapping and the role mapping (e.g. to sn).

- <READ_PRINCIPAL_NAME>

  If you use different attributes for storing user names and group/role names, you must enable read access and configure the appropriate value for the <ldapattr> attribute (e.g. sn).

Example

You want to use the fullname attribute for user names and the organame attribute for group/role names. The mappings look as follows:

```
<CN>
      <vipattr>cn</vipattr>
      <ldapattr>fullname</ldapattr>
</CN>
<LDAP_USER_NAME>
      <vipattr>userclass.name</vipattr>
      <ldapattr>fullname</ldapattr>
      <ldapread>false</ldapread>
</LDAP_USER_NAME>
<LDAP_GROUP_NAME>
      <vipattr>groupclass.name</vipattr>
      <ldapattr>organame</ldapattr>
      <ldapread>false</ldapread>
</LDAP_GROUP_NAME>
<LDAP_ROLE_NAME>
      <vipattr>roleclass.name</vipattr>
      <ldapattr>organame</ldapattr>
      <ldapread>false</ldapread>
</LDAP_ROLE_NAME>
<READ_PRINCIPAL_NAME>
      <vipattr>principal.name</vipattr>
      <ldapattr>organame</ldapattr>
      <ldapread>true</ldapread>
      <ldapwrite>false</ldapwrite>
</READ_PRINCIPAL_NAME>
```

Speeding up LDAP Requests

You can use various configuration options for speeding up LDAP requests performed by Livelink WCM Server. These include indexing functions in the LDAP servers and inverse attributes.

Using the index of the LDAP server

In some LDAP directory services, attributes can be included in an index. This speeds up searches for objects with this attribute. You should index the following WCM attributes:

- `vipFuncarea`, `vipWebsite`, and `vipSubstitute` (WCM attributes in which assignments are saved)
- `uid` and `cn` (WCM attributes from which user IDs are read)
- `member` and `uniquemember` (WCM attributes in which the users of a group or role are saved)
- `objectclass`
- In Microsoft Active Directory, `vipType` should also be indexed.

Using inverse LDAP attributes

If many groups and users are managed in the LDAP directory service, retrieving certain user settings, such as the group membership, can be time-consuming. To speed up such requests, Livelink WCM Server can use so-called *inverse LDAP attributes*.

These attributes save inverse assignments – for example, the groups/roles a user belongs to can be saved in the attribute `memberof` of the class `vipUser`. Another inverse attribute that Livelink WCM Server can use is `substituteof` for saving assignments of substitutes.

If these attributes have been mapped and set correctly and the settings/ assignments of a user are, for example, retrieved via the Admin client, the system does not search all groups to check whether the selected user belongs to them. Instead, only the attribute `memberof` is analyzed. The same method is used for reading substitutes.

Notes:

Livelink WCM Server can use these attributes to perform requests more quickly. However, these attributes must be maintained outside the WCM system.

You should only make these attributes available to Livelink WCM Server if a very high number of groups and roles are managed in your LDAP directory service and performance problems occur when user data is retrieved.

If you want to use inverse LDAP attributes, add the following attributes to the `vipUser` object class:

Table 6 – Inverse LDAP attributes for the class `vipUser`

WCM attribute	Data type (syntax)	Single value	Mapping WCM attribute → LDAP attribute
`vipMemberOf`	DN		`vipMemberOf` → `memberof`
`vipSubstitueOf`	DN		`vipSubstituteOf` → `substituteof`

These attributes must be mapped to the attributes of the LDAP directory service in the **ldapmapping.xml** file in the directory **{WCM installation directory}\config**.

Example (Active Directory):

```
...
<VIP_MEMBEROF>
      <vipattr>vipMemberOf</vipattr>
      <ldapattr>memberof</ldapattr>
</VIP_MEMBEROF>
<VIP_SUBSTITUTEOF>
      <vipattr>vipSubstituteOf</vipattr>
      <ldapattr>substituteof</ldapattr>
</VIP_SUBSTITUTEOF>
...
```

Creating a separate directory for WCM users

If the LDAP directory service manages many users with numerous attributes, of which only some users are to access the WCM system and for which only some attributes are relevant for the WCMS, we recommend that you offload these users with the attributes relevant for the WCMS to a separate directory (e.g. by means of the replication mechanism of the LDAP directory service). This speeds up access to the WCM system.

Creating the LDAP Binding Profile and the WCM Administrator

Livelink WCM Server accesses the LDAP directory service via the so-called LDAP binding profile, i.e. a connection with read and write access to the WCM-specific object classes and attributes.

For this connection, Livelink WCM Server uses the “Simple Authentication” procedure, i.e. authentication is performed by means of a password transmitted in plain text. To improve security, SSL (Secure Socket Layer) can be used for the connection to the LDAP directory service (see section “Using SSL Connections to the LDAP Server” on page 61).

When installing Livelink WCM Server, you must additionally specify a user as WCM administrator. This user must already exist in the LDAP directory service when the WCM system is installed.

Configuring the LDAP Binding Profile

Livelink WCM Server accesses the LDAP directory service via a personalized connection. This connection requires a binding profile in LDAP that can be used to handle all read and write accesses of Livelink WCM Server. During the installation of Livelink WCM Server, you specify this binding profile together with a password (see section “Setting the Parameters for the LDAP Directory Service” on page 82).

You can use an existing LDAP administrator account as the binding profile. If you do not want to do this, you must create a new binding profile in LDAP. In that case, note the following:

- For security reasons, the user should be positioned outside the namespace accessible to Livelink WCM Server. This prevents access to the properties of the binding profile via the Admin client.
- The user must have read and write access to the attributes used by the WCM system starting from the specified search node.

Setting up the WCM Administrator

During the installation, you configure a user as WCM administrator (see section “Setting up the WCM Administrator” on page 88). This user must exist in LDAP before the installation and must have the following WCM attributes:

- `vipAccess`, value = `true` (i.e. active)
- `cn`
- `mail`

- uid
- vipLanguage, value = de_DE or en_US (locale)
- userPassword
- initPassword, value = false
- trustedLogin, value = false
- For Microsoft Active Directory: vipType, value = vipUser
- For Novell eDirectory, Sun ONE Directory Server, and OpenLDAP: The value vipUser must be added to the attribute objectclass.

No specific write access rights are required for this user because unlike the LDAP binding profile, this user is a “normal” WCM user. The WCM administrator must be created in the namespace that Livelink WCM Server accesses. Otherwise, the administrator cannot log in to the WCM system.

Notes on Using Multiple LDAP Servers

Livelink WCM Server can access multiple LDAP servers. This may be necessary if a company uses different LDAP servers for managing user information or fallback LDAP servers to increase failover protection. The additional LDAP servers are integrated in the WCM system by means of additional LDAP pools. In the Admin client, you can define the order in which the LDAP servers are to be accessed. If an LDAP server is not accessible, it will be ignored for a certain period of time.

Note: When using several LDAP servers – for different Administration servers or when using fallback servers – the data between the LDAP servers must be replicated on an up-to-date basis.

Moreover, several WCM systems can access the user information of the same LDAP server. For further information, refer to section “Using User Information in Different WCM Systems” on page 58.

Using Fallback LDAP Servers

To increase failover protection, it is possible to use fallback LDAP servers which take over the tasks of the primary LDAP server if this server fails. After the installation, you must create a pool for the fallback LDAP server in the Admin client and assign this pool to the Admin server. Otherwise, the fallback server cannot be reached from the WCM system. For information on creating and assigning new pools, refer to the Livelink WCM Server Administrator Manual.

Notes:

When installing Livelink WCM Server, the primary LDAP server is specified (see section “Setting the Parameters for the LDAP Directory Service” on page 82).

When accessing an LDAP server, the WCM system does not differentiate between the primary LDAP server, an alternative LDAP server, and a fallback system. Please keep this in mind when configuring the fallback LDAP server.

Using User Information from Several LDAP Servers

If you use several LDAP servers for managing user information in your company, Livelink WCM Server can access different LDAP servers. Thus, you can, for example, grant WCM access to the group “Marketing” from LDAP server A and to the group “Sales” from LDAP server B.

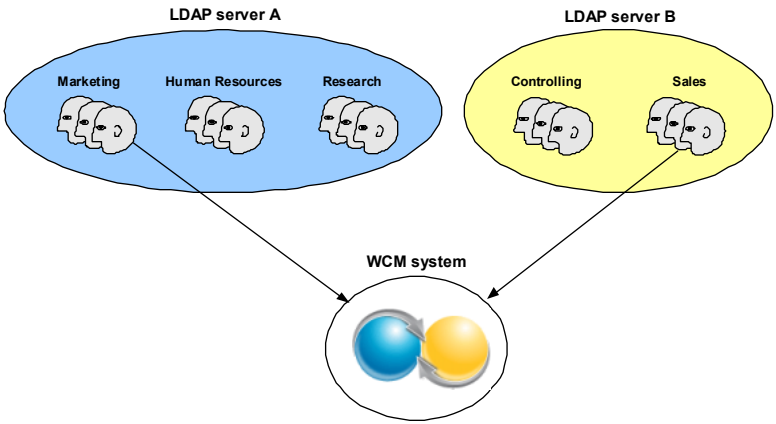


Fig. 2 – Using several LDAP servers

When installing Livelink WCM Server, you first specify the LDAP server on which the future WCM administrator is managed, e.g. LDAP server A (see sections “Setting the Parameters for the LDAP Directory Service” on page 82 and “Setting up the WCM Administrator” on page 88). After the installation, you can use the Admin client to add the WCM attributes to the users of the group “Marketing” from LDAP server A.

To enable the users of the group “Sales” from LDAP server B to access the WCM system, create a new LDAP pool with the connection parameters of this server in the Admin client and assign this pool to the Admin server. Now you can add the WCM attributes to the users of the group “Sales”, thus enabling them to access the WCM system.

Notes

- For information on how to use the Admin client to add WCM attributes to LDAP users, refer to chapter “User Administration” of the Livelink WCM Server Administrator Manual.
- The LDAP position of the principals (users, groups, and roles) is not reflected in the WCM system. The principals are represented in a flat list in the Admin client.
- The user IDs must be unique for all LDAP branches specified, as the users in the WCM system are differentiated on the basis of their ID and not on the basis of their assignment to groups or roles or their position in the LDAP tree.

  The group and role names must also be unique within the WCM system.
- When new users, groups, and roles are created, they must be clearly assigned to one LDAP server if different search nodes are used. When creating the principal, enter the correct LDAP position.

Using User Information in Different WCM Systems

It is possible to use user information from one LDAP directory service for several WCM systems. The following diagram illustrates such a scenario. The users of the group “Marketing” can access two WCM systems, whereas the group “Sales” only has access to the WCM system B.

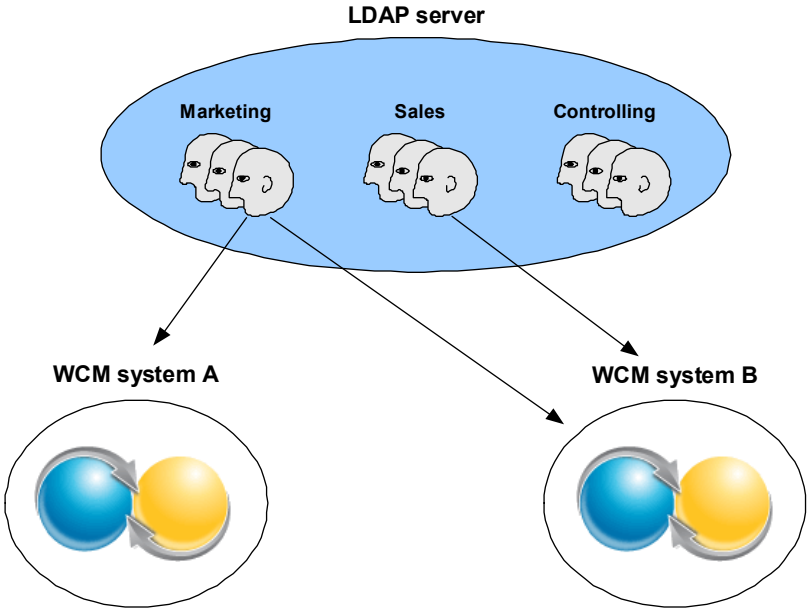


Fig. 3 – Using one LDAP server for several WCM systems

There are two ways of configuring such a scenario:

- Both WCM systems use identical LDAP object classes and attributes.

  Thus, users of the group “Marketing” have the same settings in both WCM systems. If you deny a user of this group access to the WCM system A, this user can no longer access WCM system B as the respective attribute (`vipAccess`) exists only once.

- The WCM systems use different LDAP object classes and attributes. This way, you can control the user settings separately for each WCM system.

In our example, the object classes of both WCM systems are added to the users of the group “Marketing”. Attributes that are to be controlled independently for the WCM systems are created separately for each WCM system. For example, two attributes for access to the WCM system are created: `vipAccessA` and `vipAccessB`.

The different attributes are assigned to the WCM system using the mapping of the WCM attributes to LDAP attributes (see section “Mapping WCM Attributes to LDAP Attributes” on page 42).

The following table illustrates a sample configuration:

Table 7 – Sample configuration for separate LDAP object classes and attributes

	WCM system A	WCM system B	LDAP principal
Object class	`vipUserA`	`vipUserB`	`vipUserA` and `vipUserB`
WCM access (controlled separately)	Mapping to the attribute `vipaccessA`	Mapping to the attribute `vipaccessB`	`vipaccessA =true` `vipaccessB =false`
Trusted login (equal for both WCM systems)	Mapping to the attribute `trustedLogin`	Mapping to the attribute `trustedLogin`	`trustedlogin =false`

Using SSL Connections to the LDAP Server

In order to prevent LDAP passwords from being transmitted in plain text via the network and thus increase security, the procedure SSL (Secure Socket Layer) can be used for the connection between the WCM system and the LDAP directory service. For this purpose, the LDAP server must provide an SSL port and the server certificate must exist as a file. For further information, please refer to the manufacturer of the LDAP directory service used.

The use of SSL can be enabled during the installation of Livelink WCM Server (see section “Setting the Parameters for the LDAP Directory Service” on page 82). You can also enable SSL subsequently in the settings of the respective LDAP pool (see Livelink WCM Server Administrator Manual).

For using SSL, the server certificate must be added to the WCM truststore first. This is done by means of the so-called key tool, which is called via the console. The key tool is located in the directory **{Java SDK directory}\bin**.

Using SSL during the installation

Before starting the installation of the WCM system, you must perform the following steps:

- Installation from hard disk: When copying the files from the WCM CD to your hard disk, proceed as described in the following section. In this case, the placeholder “{WCM installation directory}” stands for the directory on your hard disk in which the installation files are saved.
- Installation from the WCM CD: If you perform the installation directly from the installation CD, proceed as follows:

1. Copy the file **gauss_vip80.trust** from the directory **\keys** on the WCM CD to a temporary directory.
2. Perform the steps described in the following section.
3. Modify the installation script **install.bat** or **.sh** by replacing the expression

```
java -Xmx64m de.gauss.vip.installation.Installation
```

with

```
java -Xmx64m -Djavax.net.ssl.truststore={location of the file
  gauss_vip80.trust} de.gauss.vip.installation.Installation
```

Adding the LDAP server certificate to the WCM truststore

To add the LDAP server certificate to the WCM truststore:

1. Copy the file with the server certificate for the LDAP server to the directory **{WCM installation directory}\keys**.
2. Open the prompt.
3. Change to the WCM installation directory and then to the subdirectory **\keys**.
4. Check whether the copied server certificate can be used by the key tool. Enter the following command:

```
keytool -printcert -file {file name of the server certificate}
```

5. Check the output of the key tool. Among other things, the tool outputs the “fingerprints” of the certificate on the console.
6. If the check was successful, you can add the server certificate to the WCM truststore, e.g. by means of the following command:

```
keytool -import -alias {alias} -v -file {file name of the server
   certificate} -keypass {password} -keystore gauss_vip80.trust
   -storepass changeit
```

The following table explains the parameters of the call which must be modified according to your system. The values that you must modify are enclosed in curly brackets, e.g. `{password}`.

Table 8 – Parameters for the keytool

Parameter	Explanation
`-alias {alias}`	Name under which the server certificate is stored in the WCM truststore
`-v`	This parameter controls how detailed the output of the key tool is on the console. `-v` means verbose output.
`-file {file name of the server certificate}`	Name of the file with the server certificate
`-keypass {password}`	Password for the file with the server certificate
`-keystore {WCM truststore}`	Name of the WCM truststore saved in **{WCM installation directory}\keys** Default = `gauss_vip80.trust`
`-storepass {password}`	Password for the WCM truststore Default = `changeit`

7. Use the following command to check whether the server certificate has been installed successfully.

```
keytool -list -v -keystore gauss_vip80.trust -storepass changeit
```

The output of the server certificate must correspond to the output in step 5.

Example

```
Keystore type: jks
Keystore provider: SUN

Your keystore contains 12 entries:

..

Alias name: {alias}
Creation date: Tue Mar 28 12:52:01 GMT+01:00 2002
Entry type: trustedCertEntry

Owner: CN=Admin, OU=Development, O=Company, C=DE
Issuer: CN=Thawte Test CA Root, OU=TEST TEST TEST, O=Thawte
   Certification, ST=FOR TESTING PURPOSES ONLY, C=ZA
Serial number: 268114
Valid from: Tue Mar 05 15:30:19 GMT+01:00 2002 until: Tue Mar 26
   15:30:19 GMT+01:00 2002
Certificate fingerprints:
       MD5:  5C:82:D7:01:AF:F9:5C:25:E7:F5:EE:D0:AE:DC:3D:ED
       SHA1: 36:5A:82:3D:B9:7B:32:2A:38:57:83:02:DD:F5:36:82:C6:
             BC:13:BA
```

The server certificate in our example was added successfully to the WCM truststore.

CHAPTER 4

Installing Livelink WCM Server

You can use the installation program of Livelink WCM Server to perform different tasks. The following figure gives you an overview of the different installation options.

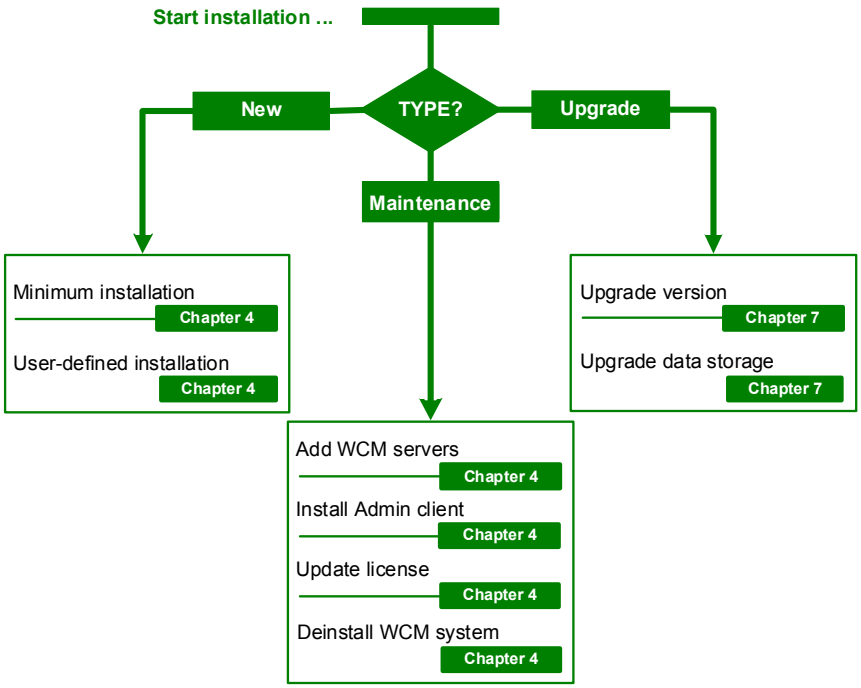


Fig. 4 – Overview of installation options

As an alternative to the installation via the graphical user interface, you can use the console (see section “Installation and Deinstallation via the Console” starting on page 126).

Moreover, you get information on starting and stopping the servers (see section “Starting and Stopping Servers” on page 151). Section “Log Files and Error Files” on page 162 contains information on the logs created during the installation or when starting/stopping the servers.

Installing a WCM System (Minimum System)

The WCM system is installed by means of the WCM installation program. A wizard guides you through the installation process and requests all the necessary information.

This section describes the installation of a minimum system. For information on the architecture of WCM systems, refer to chapter “Concepts” of the Livelink WCM Server Administrator Manual.

Notes on installing

- For the installation, valid licenses for all WCM components are required.
- If you want to set up the servers as Windows services, you need local administrator rights.
- Under UNIX, you should not install and operate the servers with the uid “0” or “root”.
- Before installing the servers under UNIX, you must create a user for the servers (e.g. “wcmuser”).

- For working with the Content client, the Content client (Classic), and InSite Editing, you require at least one Content server running in the context of a JSP engine or as a web application in an application server.
- All the information required for the WCM system is entered during the installation. In certain cases, it is necessary to make some settings in the **defaults.xml** file before installation. This file is located in the **\installation** directory on the WCM CD and contains the default settings for the installation.

  To make changes in this file, for example regarding the mapping of LDAP attributes, copy the file to a local drive. When you start the installation, you can specify the path to the **defaults.xml** file as a parameter.
- Already during the installation of Livelink WCM Server, you can use the procedure SSL (Secure Socket Layer) for the connection from the WCM system to the LDAP directory service. For this purpose, certain steps must be performed before the installation (see section “Using SSL during the installation” on page 61).

Starting the Installation

The installation program is started differently under Windows and UNIX:

- **Windows:** Open the MS-DOS prompt, change to the CD-ROM drive, then start the installation program, specifying the path to your Java 2 SDK directory as a parameter. The path to the **defaults.xml** file (including file name) only needs to be specified if a locally stored file is to be used instead of the standard file on the WCM CD.

```
install.bat {SDK installation directory}\bin
   {path to the defaults.xml file}
```

- **UNIX:** A JAR archive is supplied on the WCM CD. Copy this archive to your server and extract it to a temporary directory. Change to this directory and call the installation script with the path to your Java 2 SDK directory and the path to the **defaults.xml** file (including file name) as parameters.

```
sh ./install.sh [-jdk {SDK installation directory}/bin]
   [-defaults {path to defaults.xml file}]
```

  If you call the installation script without any parameter, you will be asked to enter the path to your Java 2 SDK directory (**{SDK installation directory}/bin/**) and to the **defaults.xml** file.

After you entered the path, the graphical user interface of the installation program opens.

A wizard guides you through the installation process. Follow the instructions displayed. The following description contains additional information going beyond the instructions of the wizard.

Selecting the Installation Directory

When installing a new WCM system, enter a local directory that does not contain any WCM components.

If you have already installed a WCM system and want to add new servers, there are two possibilities:

- If the new server is to be installed in the same directory as the existing WCM system, enter the WCM installation directory.
- If the new server is to be installed in a different directory, enter this directory.

Note: Please note that under UNIX the installation directory may not be a symbolic link.

Confirm by clicking the *Next* button.

Selecting the Type of Installation

In this dialog box, you specify the component to be installed.

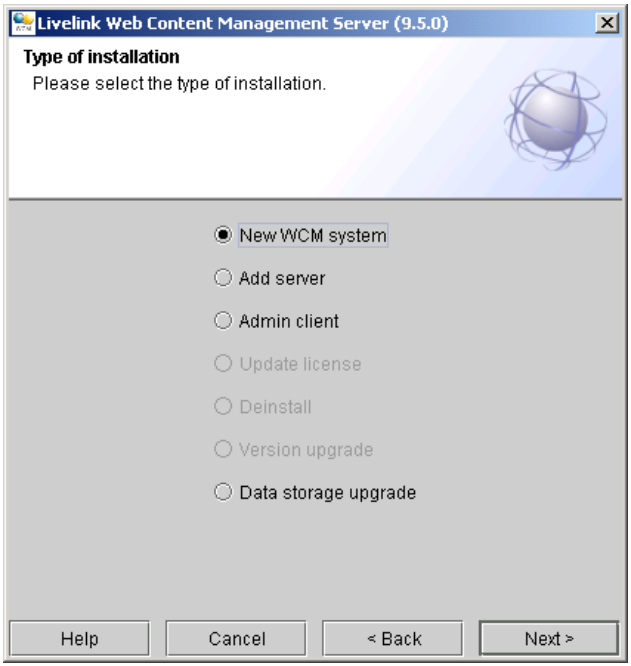


Fig. 5 – Installation options

Note: The *New WCM system* radio is automatically dimmed, if you selected an installation directory which already contains a WCM system.

- *New WCM system:* Click this radio button to set up a new WCM system.
- *Add server*, see section “Adding New Servers” on page 110
- *Admin client*, see section “Installing the Admin Client” on page 113
- *Update license*, see section “Updating the License” on page 115
- *Deinstall*, see section “Deinstalling the WCM System” on page 121
- *Version upgrade*, see section “Upgrading the Version” on page 214
- *Data storage upgrade*, see section “Upgrading the Data Storage” on page 217

Click the desired radio button and then the *Next* button.

Specifying the License Management Options

In this dialog box, you specify the options for the license management.

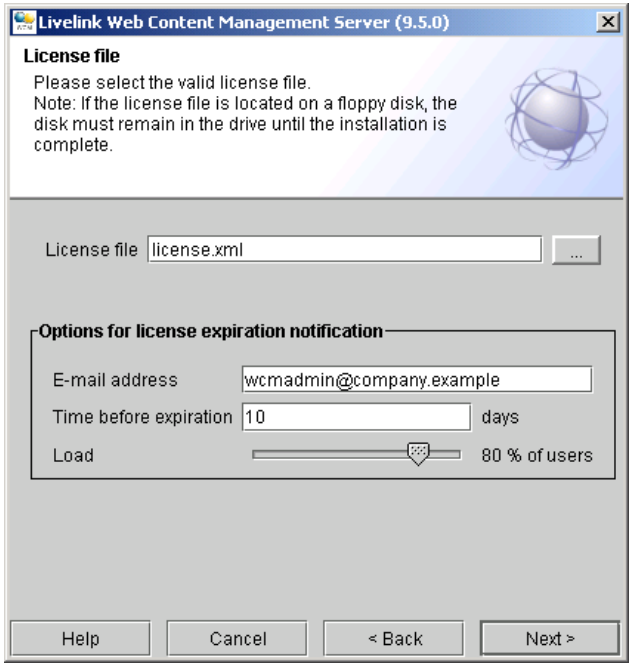


Fig. 6 – Specifying the license management options

- *License file:* Click the button ... to select the path to the supplied license file **license.xml**.
- *E-mail address:* e-mail address of the person who is to be informed by e-mail if the WCM licenses expire or the number of used licenses exceeds a specified limit
- *Time before expiration:* number of days before the licenses expire. If this point in time is reached, a message is automatically sent to the e-mail address specified.
- *Load ... % of users:* percentage of licenses used by active users. If this percentage is exceeded, a message is automatically sent to the e-mail address specified.

Notes:

After the installation, these settings can be changed in the Admin client via *Configuration* → *Utilities* → *License expiration notification*.

You can check the status of your WCM licenses at any time via *System administration* → *Licenses*.

Confirm by clicking the *Next* button.

Setting the Parameters for the New Administration Server

In this dialog box, you define the parameters for the new Administration server.

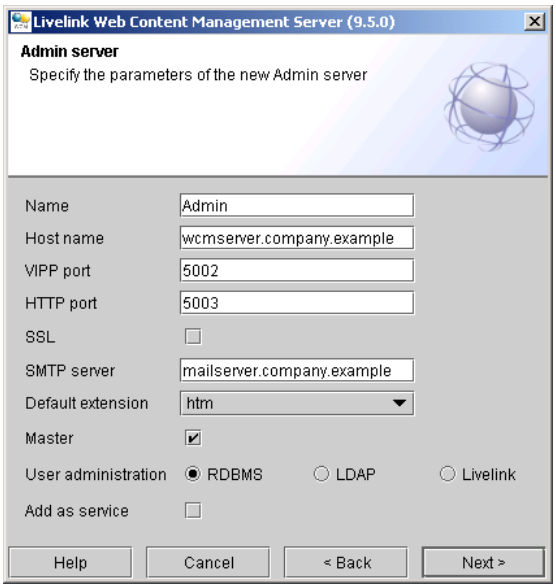


Fig. 7 – Parameters of the Administration server

- *Name:* name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted:
  - a-z, A-Z, 0-9 (ASCII 7 Bit)
  - . (dot), _ (underline), and - (hyphen)
- *Host name:* fully qualified name of the computer on which this server is to be installed

  By default, the program attempts to determine the name of the current computer. If this is not possible, the default entry “localhost” is used. You should change this entry, however, as it causes problems in distributed systems.
- *VIPP port* and *HTTP port:* server ports for connections via the protocols VIPP and HTTP. The installation program creates a pool for the connections to this server which gets the same name as the server.

  Notes:

  Make sure that the port numbers are not used by other applications. If firewalls exist between the individual servers or between client and server, the relevant port addresses must be enabled.

  After installation, you can use the Admin client to specify additional parameters for the communication via the respective protocol (*Configuration* view → *Pools* → *WCM* → *{pool name}* → *WCM connection* tab).
- *SSL:* Select the *SSL* check box to enable secure connections (SSL=Secure Socket Layer) for the communication between the servers.
- *SMTP server:* fully qualified name of the computer hosting the SMTP server. After the installation, you can change the address of the SMTP server in the Admin client (server settings, *Miscellaneous* tab).

- *Default extension:* Select the standard file name extension for generated pages. The deployment systems generate these pages from the WCM objects and store them in the local file system of the servers.
- *Master:* Select this check box to set up the Administration server as a master Admin server. If you want to install a proxy Admin server, deselect this check box.
- *User administration:* Select whether the Administration server is to store user data in a relational database management system, in an LDAP directory service, or in Livelink.

  Note: If the user data of the Livelink system is synchronized with an LDAP directory service, we recommend that you click the *LDAP* radio button here.

- *Add as service:* Select this check box to set up a Windows service for the Administration server (for automatic server startup and shutdown).

  Note: By means of the **install_{server name}.bat** scripts in the directory **{WCM installation directory}\installation\service**, you can set up services for the servers after installation.

Confirm your entries by clicking the *Next* button.

Setting RDBMS Parameters

If you have clicked the *RDBMS* radio button for *User administration* when configuring the Admin server, you must specify the parameters for the connection to the database in the next dialog box.

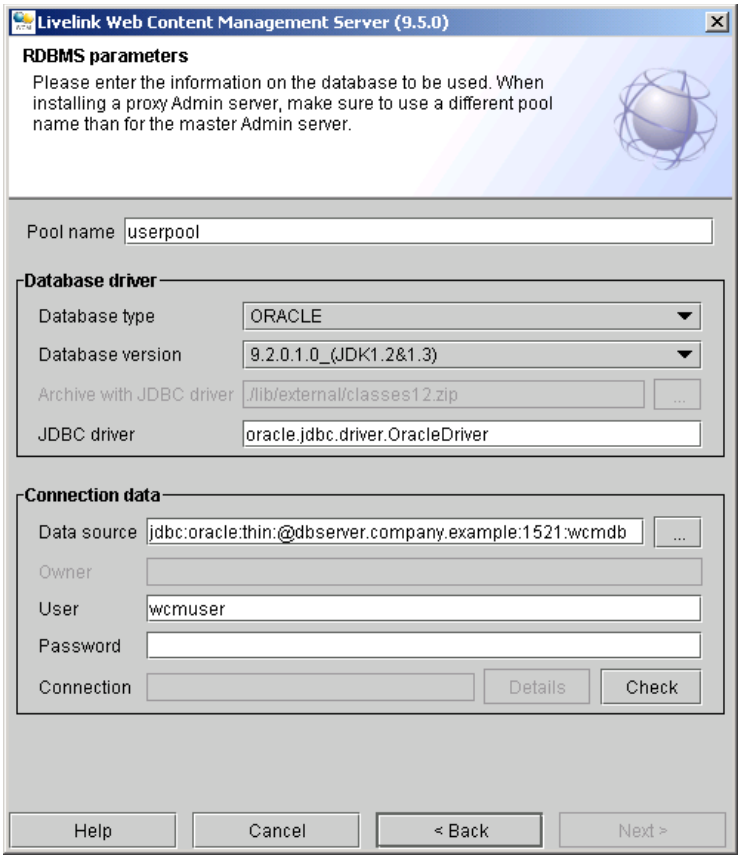


Fig. 8 – Parameters of the database to be used

Pool name: name of the JDBC pool created for the connections to the RDBMS

> **Important!** When installing a proxy Admin server, make sure to use a different name for this pool than for the pool of the master Admin server.

***Database driver* section**

- *Database type:* type of the RDBMS used. You can choose between
  - ORACLE
  - MS SQL Server 2000
- *Database version* (Oracle only): version of the Oracle RDBMS used. If the version you use is not available in the list, select the entry “Select archive manually”.
- *Archive with JDBC driver:* The driver for certain versions of the Oracle RDBMS is supplied with Livelink WCM Server. This field only becomes active if you use a different RDBMS or a different version of the Oracle RDBMS. In this case, select the file(s) with the JDBC drivers for the database used.

  The files are written to the field with their complete path. Several paths are separated by semicolons.

  The installation program copies these file(s) to the directory **{WCM installation directory}\external_lib**. This is done on all servers. For information on JDBC drivers for databases, contact the respective manufacturer.

  Notes

  - Always use the JDBC driver matching the current version of the database and the Java 2 SDK you use. When the version is updated (e.g. by means of a fixpack), the version of the JDBC driver may also change. When using MS SQL Server 2000, replace the respective files in the directory **{WCM installation directory}\external_lib** and restart the respective servers.

- If you have installed an Oracle patch containing a new JDBC driver, replace the driver file (e.g. **classes12.zip**) in the directory **{WCM installation directory}\external_lib** with the current file after installing the patch.
- If the name of the driver class changes subsequently, a new pool must be configured for the connection to the database and assigned to the respective servers.

- *JDBC driver:* driver class for the RDBMS used

  If you have selected Oracle as the database type, the name of the driver class will be set automatically.

  If you use a different RDBMS, enter the name of the driver class. For example, if you use the Microsoft JDBC driver for the RDBMS MS SQL Server 2000, enter `com.microsoft.jdbc.sqlserver.SQLServerDriver`. For further information, refer to the documentation of the JDBC driver used.

***Connection data* section**

- *Data source:* full name of the data source

  If you use the Oracle RDBMS, click the button ... to open the *JDBC data source* dialog box in which you can set the exact parameters for the connection to the RDBMS.

  - *Database host:* fully qualified host name or IP address of the computer hosting the database
  - *Database port:* connection port of the database. The default value for the selected database type is entered automatically.
  - *Database name:* name of the database

If you use a different RDBMS, enter the URL for accessing the RDBMS in the field. The format of the URL depends on the JDBC driver used. For the Microsoft JDBC driver for the RDBMS MS SQL Server 2000, the following must be entered: `jdbc:microsoft:sqlserver://{database host}:1433;SelectMethod=cursor;DatabaseName={database name}`. For further information, refer to the documentation of the JDBC driver used.

- *Owner* (for SQL Server only): ID of the user who created the database objects used by Livelink WCM Server in the RDBMS
- *User:* user ID for access to the RDBMS. When using the Oracle RDBMS, this user ID may not contain hyphens.
- *Password:* password for access to the RDBMS

After this, click the *Check* button to test the database connection. If all entries have been made correctly, *OK* is displayed. In this case, you can click the *Details* button to display information on the RDBMS and JDBC driver used.

Confirm by clicking the *Next* button.

Specifying the Parameters of the Livelink System

If you have clicked the *Livelink* radio button for *User administration* when configuring the Admin server, you must specify the parameters for the connection to the Livelink system in the next dialog box.

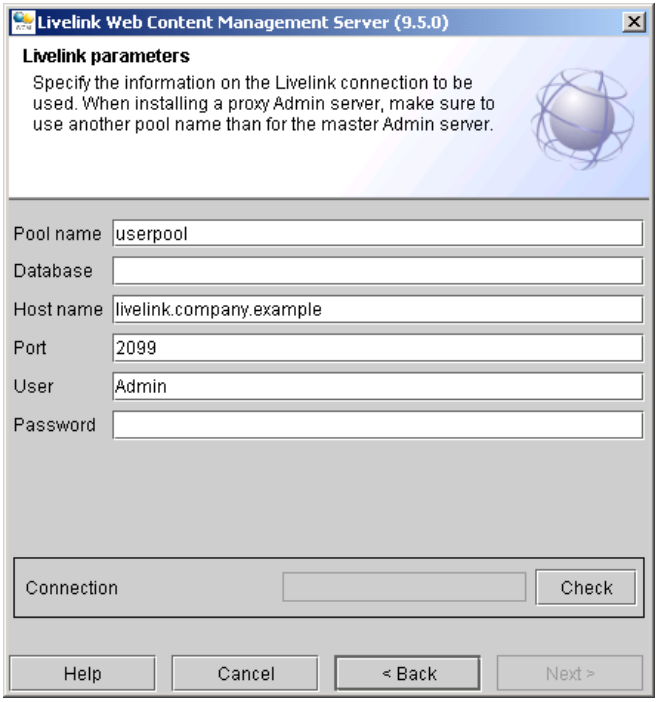


Fig. 9 – Parameters of the Livelink system to be used

- *Pool name:* name of the Livelink pool created for the connections to the Livelink system

  Important! When installing a proxy Admin server, make sure to use a different name for this pool than for the pool of the master Admin server.

- *Database:* logical name of the Livelink database to be used. Normally, this field remains empty, i.e. the database configured as the default database in Livelink will be used.
- *Host name:* fully qualified name of the computer hosting the Livelink server

- *Port:* connection port on the Livelink server
- *User:* user ID for access to the Livelink server. The user must be allowed to add, modify, and delete users and groups in the Livelink system.
- *Password:* password for access to the Livelink server

After this, click the *Check* button to test the connection to the Livelink system. If all entries have been made correctly, *OK* is displayed.

Setting the Parameters for the LDAP Directory Service

If you have clicked the *LDAP* radio button for *User administration* when configuring the Admin server, you must specify the parameters for the connection to the LDAP directory service in the next dialog box.

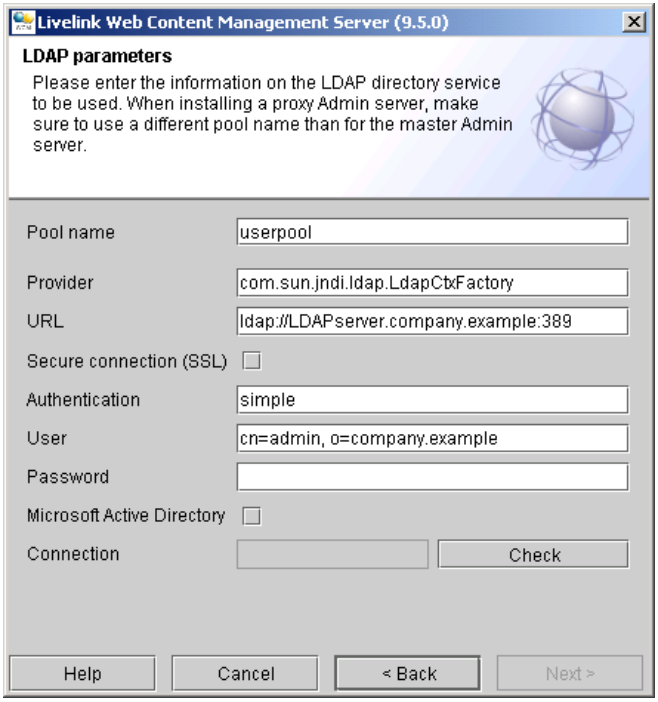


Fig. 10 – Parameters of the LDAP directory service to be used

Note: If the users who are to access the WCM system are managed in different LDAP servers, enter the LDAP server on which the future WCM administrator is stored. After the installation, you can use the Admin client to create additional LDAP pools for accessing further LDAP servers or fallback LDAP servers.

- *Pool name:* name of the LDAP pool created for the connections to the LDAP server

> **Important!** When installing a proxy Admin server, make sure to use a different name for this pool than for the pool of the master Admin server.

- *Provider:* driver used for the LDAP server. The Java class stated here must be accessible via the class path set in the system. The default class `com.sun.jndi.ldap.LdapCtxFactory` corresponds to the LDAP standard and can be used for LDAP directory servers of various manufacturers.
- *URL:* address of the LDAP server consisting of the protocol ldap://, the name of the LDAP server, and the port for the LDAP connections (default = 389)
  Example: “ldap://LDAPserver.company.example:389”
- *Secure connection (SSL):* Select this check box to enable secure connections (Secure Socket Layer) for the communication with the LDAP server.

  If you want to use SSL, the LDAP server certificate must be added to the WCM truststore before the installation (see section “Using SSL Connections to the LDAP Server” on page 61).
- *Authentication:* authentication method for logging in to the LDAP server. The method “simple” must be entered here.
- *User:* user ID of the binding profile which is used to access the LDAP directory service. Enter the “distinguished name” of the user.

  Please note the information on the binding profile in section“Configuring the LDAP Binding Profile” on page 54.
- *Password:* password of the binding profile

> **Important!** If the password is changed in the LDAP directory service, the WCM system can no longer access the LDAP directory service.

- *Microsoft Active Directory:* Select this check box if you use Microsoft Active Directory.

After this, click the *Check* button to test the connection to the LDAP server. If all entries have been made correctly, *OK* is displayed.

Confirm by clicking the *Next* button.

Specifying WCM-Specific LDAP Parameters

Livelink WCM Server requires its own object classes in the LDAP directory service. These object classes must be added to LDAP before the installation (see section “Creating Additional Object Classes and Attributes” starting on page 38). In this dialog box, enter the names of these object classes.

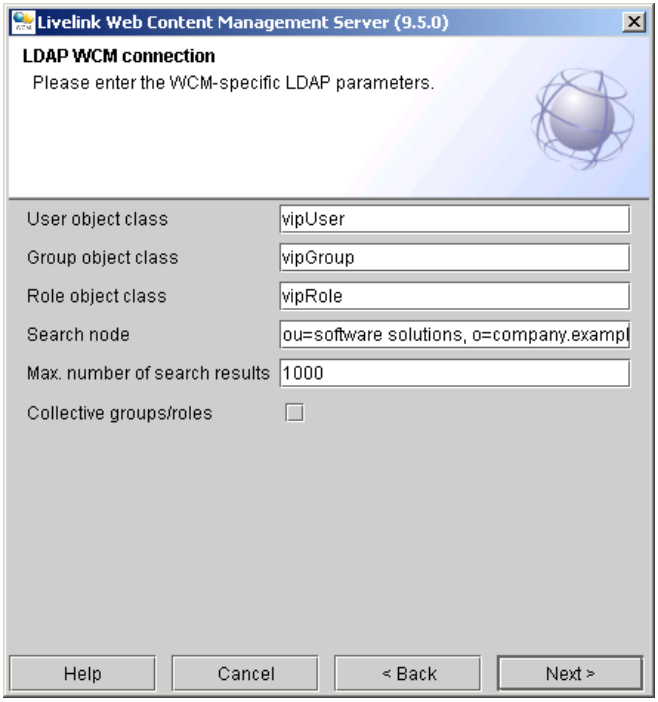


Fig. 11 – WCM-specific LDAP parameters

- *User object class:*object class for saving the user information. The default name is `vipUser` (`user` for Microsoft Active Directory).
- *Group object class:*object class for saving the group information. The default name is `vipGroup` (`group` for Microsoft Active Directory).
- *Role object class:*object class for saving the role information. The default name is `vipRole` (`group` for Microsoft Active Directory).
- *Search node:* root name context (base DN). Starting from this node, the WCM system has read and write access to the LDAP namespace. The search node may, for example, be composed of the organization (o) and the organizational unit (ou).

Example: “o=company.example, ou=marketing”
Livelink WCM Server can access all entries in the namespaces in and below “marketing”.

- *Max. number of search results:* maximum number of results Livelink WCM Server retrieves when searching an LDAP directory service. Depending on the product used, you can configure in the LDAP server how many search results are returned. The maximum number of retrieved search results configured for Livelink WCM Server should be less than or equal to the maximum number of returned search results set in the LDAP server.

  If you do not specify a value here, the default value 1000 will be used.

  Example: In the LDAP directory service Microsoft Active Directory, the maximum number of search results returned is controlled by the parameter `MaxPageSize`. This parameter belongs to the attribute `LDAPAdminLimits` of the Active Directory Query Policies. The Query Policies can be edited by means of the tool **Ntdsutil.exe**, for example. The Query Policies can also be accessed via the LDAP configuration context, which might look as follows: `CN=Default Query Policy,CN=Query-Policies,CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration ,DC=win2000,DC=en`.

- *Collective groups/roles:* Select this check box to determine that implicit assignments of users to groups and roles are to be considered in the WCM system (see section “Concept of Collective Groups and Collective Roles” on page 36).

> **Note:** The setting that you make here is used as global setting for the entire LDAP pool. If you have set up several LDAP pools in your WCM system, make sure that this setting is identical for pools whose LDAP contexts overlap.

Confirm by clicking the *Next* button.

Setting up the WCM Administrator

Enter the user ID and password of the WCM administrator in this dialog box. The specified user is initially granted full access to the Admin client. In the Admin client, you can subsequently grant administration rights to other users.

Depending on the user administration method (see section “Setting the Parameters for the New Administration Server” on page 74), this user must meet the following requirements:

- *RDBMS:* Enter a user ID which does not exist in the database yet.
- *LDAP:* Enter a user ID which already exists in the LDAP directory service. Make sure that you enter a different user than for the LDAP binding profile (see section “Creating the LDAP Binding Profile and the WCM Administrator” starting on page 53).
- *Livelink:* Enter a user ID which already exists in Livelink. The user must be allowed to add, modify, and delete users and groups in the Livelink system.

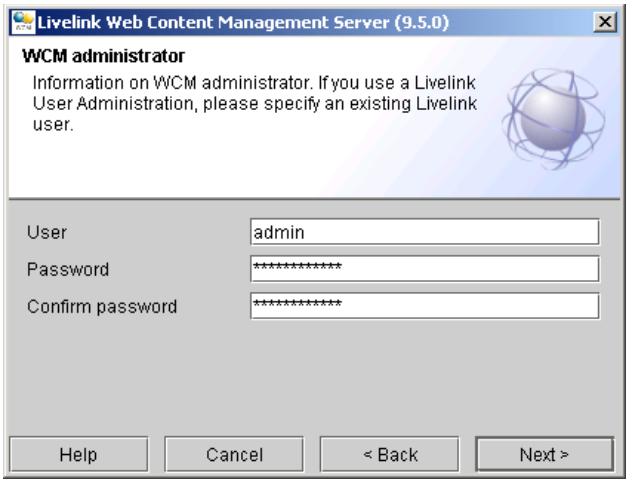


Fig. 12 – Setting up the WCM administrator

If you manage the user data in a database, enter the administrator password in the remaining fields. If you use an LDAP server or a Livelink user administration, you do not need to enter a password.

Confirm by clicking the *Next* button.

Complete the Installation or Add Servers?

In the next dialog box, you can decide whether you want to finish the installation process after installing the Administration server, or whether you want to add additional servers to the system.

- *Install the Admin server:* If you want to install the system with the settings made so far, click this radio button.

  In this case, only the Administration server is set up. You can add more servers by means of the installation program later.

  The next dialog box displays the components to be installed. To start the installation, confirm your entries by clicking the *Finish* button.

- *Add additional servers to the WCM system*: Normally, more components are installed during the installation process. Leave the default option activated and continue by clicking the *Next* button.

Selecting Option for Server Installation

For the installation of additional servers, you must specify how your WCM system is to be set up.

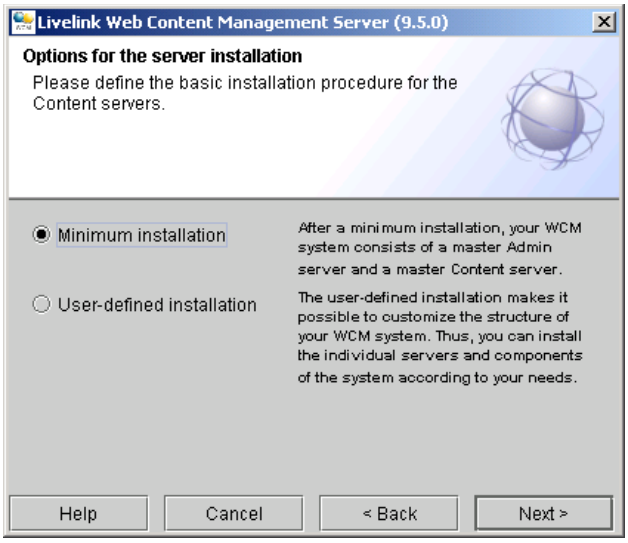

Livelink Web Content Management Server (9.5.0)
Options for the server installation
Please define the basic installation procedure for the Content servers.
Minimum installation
After a minimum installation, your WCM system consists of a master Admin server and a master Content server.
User-defined installation
The user-defined installation makes it possible to customize the structure of your WCM system. Thus, you can install the individual servers and components of the system according to your needs.
Help
Cancel
< Back
Next >

Fig. 13 – Installation variants

> **Note:** For general information on the architecture of a WCM system, refer to the Livelink WCM Server Administrator Manual (chapter “Concepts”).

Click the desired radio button and then the *Next* button.

Configuring Servers

The dialog box for configuring the Content servers opens. This dialog box contains a tab for the master Content server.

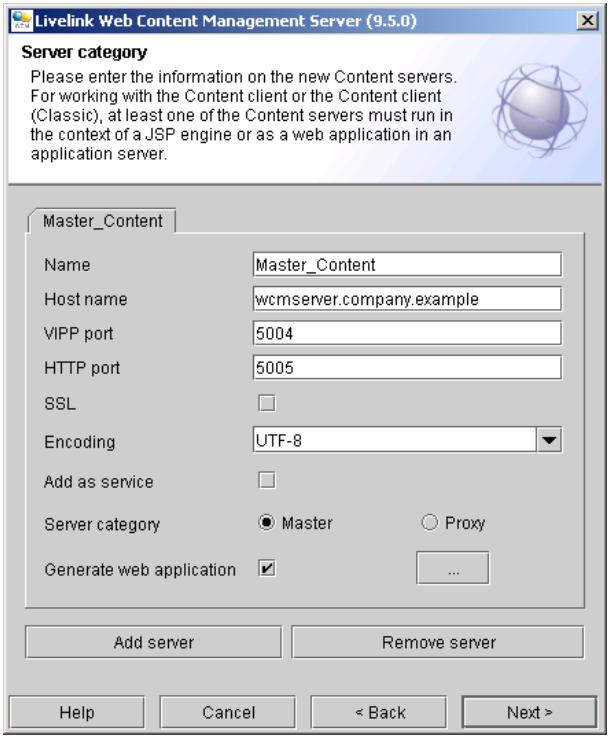


Fig. 14 – Dialog box for configuring the servers

Notes:

For editing the WCM objects by means of the Content client, you require at least one Content server running in the context of a JSP engine or as a web application in an application server.

If you set up two or more Content servers running in the JSP engine on the same computer, you must use different instances of your JSP engine for executing the servers. After the installation of Livelink WCM Server, modify the default URL in the scripts for starting the respective Content servers according to the configuration of the JSP engine used (see section “Starting a Content Server in the Application Server” on page 159).

Make the required settings for the server:

- *Name:* name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted:
  - a-z, A-Z, 0-9 (ASCII 7 Bit)
  - . (dot), _ (underline), and - (hyphen)
- *Host name:* fully qualified name of the computer on which this server is to be installed

  By default, the program attempts to determine the name of the current computer. If this is not possible, the default entry “localhost” is used. You should change this entry, however, as it causes problems in distributed systems.
- *VIPP port* and *HTTP port:* server ports for connections via the protocols VIPP and HTTP. The installation program creates a pool for the connections to this server which gets the same name as the server.

> **Notes:**
>
> Make sure that the port numbers are not used by other applications. If firewalls exist between the individual servers or between client and server, the relevant port addresses must be enabled.
>
> After installation, you can use the Admin client to specify additional parameters for the communication via the respective protocol (*Configuration* view → *Pools* → *WCM* → *{pool name}* → *WCM connection* tab).

- *SSL:* Select the *SSL* check box to enable secure connections (SSL=Secure Socket Layer) for the communication between the servers.
- *Encoding* (only for Content servers running in the JSP engine/application server): Select an encoding for the Content server and thus for the Content client.

  If you use UTF-8 encoded pages (Unicode) on your website, select *UTF-8*. If you do not need Unicode support, select *ISO-8859-1*.

  > **Note:** The encoding must be set consistently for all components of the WCM system. Please also refer to the notes on using Unicode in chapter “Concepts” of the Livelink WCM Server Administrator Manual.

- *Add as service* (not for Content servers running in the JSP engine/application server): Select this check box to set up a Windows service for the server (for automatic server startup and shutdown).

  > **Note:** By means of the **install_{server name}.bat** scripts in the directory **{WCM installation directory}\installation\service**, you can set up services for the servers after installation.

- *Server category:* The server category is suggested automatically.

  For general information on server categories, refer to the Livelink WCM Server Administrator Manual.

- *Generate web application:* For working with the Content client or the Content client (Classic), at least one Content server must run in the context of a JSP engine or as a web application in an application server.

  If you want to generate a web application for the server, select this check box and click the button ... . For a description of the parameters to be configured, refer to the following section.

You have now entered all the information required for a minimum installation. Do one of the following:

- To add another server, click the *Add server* button. A new tab opens on which you can enter the parameters required for this server.
- If you want to remove the entries for a server, go to the appropriate tab and click the *Remove server* button.
- When you have completed all server settings, click the *Next* button.

Generating a Web Application

If you select the *Generate web application* check box and click the button ..., the *Generate web application* dialog box opens.

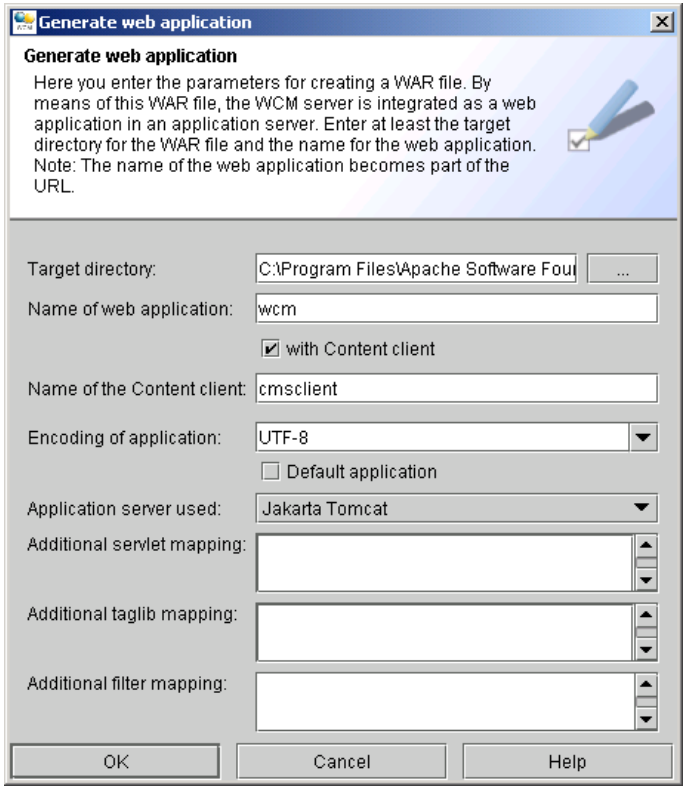

Generate web application
Generate web application
Here you enter the parameters for creating a WAR file. By means of this WAR file, the WCM server is integrated as a web application in an application server. Enter at least the target directory for the WAR file and the name for the web application. Note: The name of the web application becomes part of the URL.
Target directory:
C:\Program Files\Apache Software Foul
...
Name of web application:
wcm
with Content client
Name of the Content client:
cmsclient
Encoding of application:
UTF-8
Default application
Application server used:
Jakarta Tomcat
Additional servlet mapping:
Additional taglib mapping:
Additional filter mapping:
OK
Cancel
Help

Fig. 15 – Generating a web application for a Content server

Make the required settings:

- *Target directory:* directory for saving the generated WAR file. Mostly, this is the web application directory of the application server.
- *Name of web application:* name of the created web application. This name determines the name of the WAR file and becomes part of the URL used to call the web application in the application server.

Example: You generate a web application for the master Content server. As the name of the web application, you enter “wcm”. Accordingly, the generated WAR file is called **wcm.war**. The string “wcm” is added to the base URL of the application server. Thus, the web application for the master Content server is called via the URL “http://wcmserver.company.example/wcm”.

- *with Content client:* Select this check box to include the Content client in the web application.

  Like the name of the web application, the name of the Content client is added to the URL. Thus, it can be called via the address “http://wcmserver.company.example/wcm/cmsclient”, for example.

  Note: Both the Content client and its predecessor, the Content client (Classic), are supplied with Livelink WCM Server. If you select the check box *with Content client*, both clients are integrated in the web application. For the Content client (Classic), the name “htmlclient” is used automatically. The Content client (Classic) can then be called, for example, via the address “http://wcmserver.company.example/wcm/htmlclient”.

- *Encoding of application:* Select the encoding for the generated web application. If your website is set to UTF-8 (Unicode), click “UTF-8” in the drop-down list. Unicode is required for displaying Eastern European and Asian languages. For Western European languages, “ISO-8859-1” encoding (corresponds to Latin-1) is sufficient. The application server must be configured accordingly.

- *Default application:* If you select this check box, the system presupposes that the generated web application is defined as the default application of the application server. The default application is addressed via the root URL of the application server, i.e. the URL does not contain the application name. In this case, the Content client would be called directly via “http://wcmserver.company.example/cmsclient”. To define the generated web application as the

default application, modify the application's configuration in the application server accordingly.

- *Application server used:* Click the application server used in the drop-down list. The generated WAR file is modified to reflect the requirements of the respective product.

  Instead of a product, you can also click a servlet API standard in the list: 2.2 or 2.3. In this case, the WAR file will be generated according to the Java Servlet Specification.

- *Additional servlet mapping, taglib mapping, and filter mapping:* The servlets, tag libraries, and filter servlets used in Livelink WCM Server are mapped automatically. The Java classes used by the servlets are mapped to URLs. If tag libraries are used, the path to the TLD files used is specified via the mapping.

  If you use additional servlets, tag libraries, or filter servlets that should also be available in the generated web application, enter the respective mapping here.

  Example of a servlet mapping

```
<servlet>
   <servlet-name>MyServlet</servlet-name>
   <servlet-class>com.company.MyServlet</servlet-class>
</servlet>
<servlet-mapping>
   <servlet-name>MyServlet</servlet-name>
   <url-pattern>/servlet/MyServletAlias</url-pattern>
</servlet-mapping>
```

  Example of a taglib mapping

```
<taglib>
   <taglib-uri>
      myTaglet.tld
   </taglib-uri>
   <taglib-location>
```

```
      /WEB-INF/tld/myTaglet.tld
   </taglib-location>
</taglib>
```

Example of a filter mapping

```
<filter>
   <filter-name>MyFilter</filter-name>
   <filter-class> com.company.MyFilter</filter-class>
</filter>
<filter-mapping>
   <filter-name>MyFilter</filter-name>
   <url-pattern>/filter/MyFilterAlias</url-pattern>
</filter-mapping>
```

Configuring Search Servers

If you have a license for Content Miner or the Search Server Connector for Lucene, the dialog box for installing Search servers opens. This dialog box contains three tabs: *Content Miner*, *Lucene*, and *Livelink*. On these tabs, you can add the desired number of Search servers.

Tab *Content Miner*

> **Note:** For general information on the concepts of Content Miner, refer to the Content Miner Manual.

Click the *Add Search server* button. This opens a tab on which you can make the settings for the Content Miner Search server.

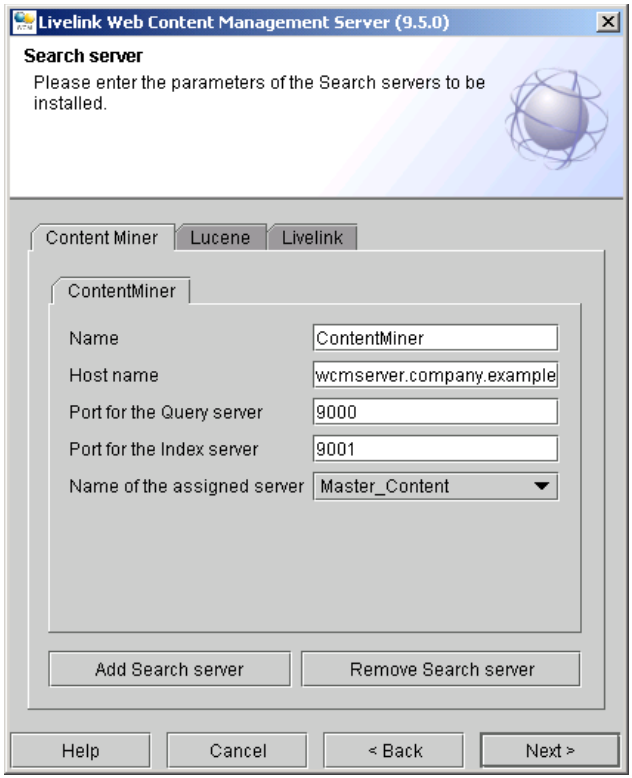


Fig. 16 – Installing a Content Miner Search server

- *Name:* name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted:
  - a-z, A-Z, 0-9 (ASCII 7 Bit)
  - . (dot), _ (underline), and - (hyphen)
- *Host name:* fully qualified name of the computer on which this server is to be installed

- *Port for the Query server:* connection port of the Query server

  The default setting is 9000. Make sure that the port is not used by any other applications or Search servers.

- *Port for the Index server:* connection port of the Index server

  The default setting is 9001. Here too, make sure that the port is not used by any other applications or Search servers.

- *Name of the assigned server:* name of a Content server that already exists or is to be installed and to which you want to assign this Search server

  Notes:

  You cannot change this assignment subsequently. To assign a Search server to a different Content server, you must first deinstall the Search server, then reinstall it, and assign it to the desired server.

  If you want the Search server to start and stop automatically with the assigned Content server and if you want to be able to start and stop the Search server via the Admin client, the Content server and the Search server must be installed on the same host (see section “Starting Search Servers” on page 160).

To add another Search server, click the *Add Search server* button again. This adds a new tab on which you can configure the next Search server.

If you want to remove the entries for a Search server, go to the appropriate tab and click the *Remove Search server* button. This will delete all the settings.

Lucene tab

> **Note:** For general information on the concepts of Lucene, refer to the Search Server Connector for Lucene Manual.

Click the *Add Search server* button. This opens a tab on which you can make the settings for the Lucene Search server.

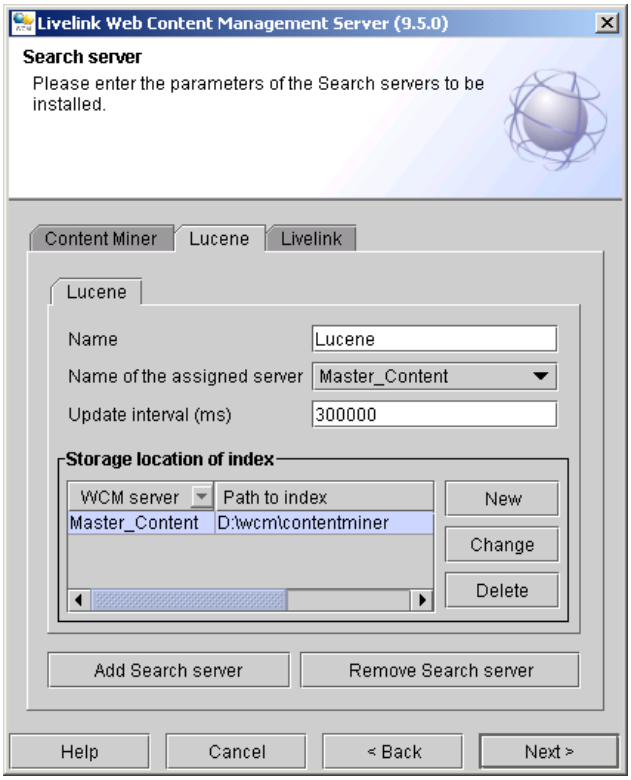


Fig. 17 – Installing a Lucene Search server

- *Name:* name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted:
  - a-z, A-Z, 0-9 (ASCII 7 Bit)
  - . (dot), _ (underline), and - (hyphen)
- *Name of the assigned server:* name of a Content server that already exists or is to be installed and to which you want to assign this Search server

> **Note:** Lucene Search servers must be installed in the same directory as the assigned Content server. This list contains only those servers that meet this condition.

- *Update interval (ms):* interval in milliseconds after which all collections will be reloaded for read access
- *Storage location of index:* In this section, you specify the path to the directory in the file system where the index is saved. This storage location must be available for all servers.

> **Note:** The storage location is set for each server individually. The individual paths must point to the same directory for all servers. If a server is located on a different host computer, the directory must be mounted on both host computers.

  - *New:* To set a new path to the index, click the *New* button. A dialog box opens. Here you can select a server and enter the path to the index.
  - *Change:* To modify the entry for a server, mark the respective server and click the *Change* button.
  - *Delete:* To remove an entry, mark the respective server and click the *Delete* button.

To add another Search server, click the *Add Search server* button again. This adds a new tab on which you can configure the next Search server.

If you want to remove the entries for a Search server, go to the appropriate tab and click the *Remove Search server* button. This will delete all the settings.

Tab *Livelink*

> **Note:** For information on integrating and using Livelink Search servers in a WCM system, refer to the Livelink Integration Manual.

Click the *Add Search server* button. This opens a tab on which you can make the settings for the Livelink Search server.

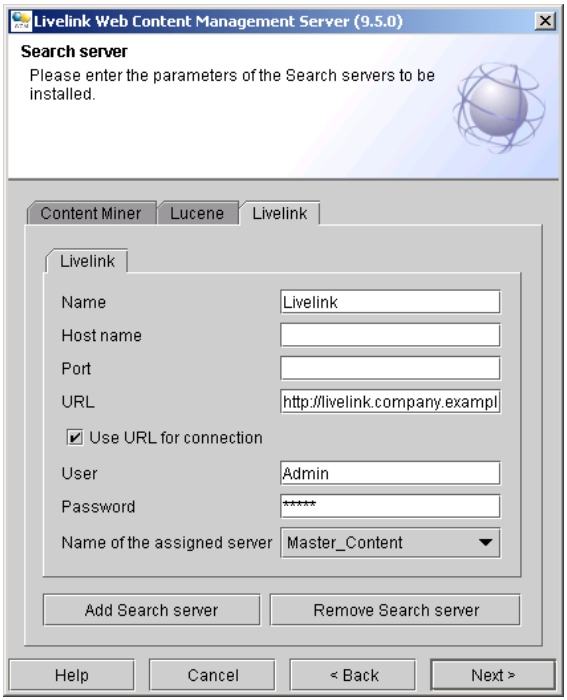


Fig. 18 – Installing a Livelink Search server

- *Name:* name of the server. You have a free choice of name, but it must be unique within the WCM system. The following characters are permitted:
  - a-z, A-Z, 0-9 (ASCII 7 Bit)
  - . (dot), _ (underline) and - (hyphen)
- *Host name:* fully qualified name of the computer hosting the Livelink server which is to perform the indexing and to process the search requests
- *Port:* connection port on the Livelink server
- *URL:* URL to the Livelink server which is to perform the indexing and to process the search requests. The URL is of type http://livelink.company.example/livelink/livelink.exe.
- *Use URL for connection:* Select this check box if you want to use the Livelink server's URL for the connection. In this case, the entries in the fields *Host name* and *Port* will be ignored.
- *User:* user for processing the search requests in the Livelink system. The user must have read access to the slices (collections).

  Notes:

  For the Livelink search, you can define which index attributes are to be queryable and/or displayable. These settings are user-specific. For this reason, we recommend that you configure a special user for the Livelink search. Enter the information of this user here.

  For information on permissions for the Livelink search, refer to the Livelink Integration Manual (chapter “Livelink Search in the WCM System”).

- *Password:* password for processing search requests in the Livelink system

- *Name of the assigned server:* name of a Content server that already exists or is to be installed and to which you want to assign this Search server

To add another Search server, click the *Add Search server* button again. This adds a new tab on which you can configure the next Search server.

If you want to remove the entries for a Search server, go to the appropriate tab and click the *Remove Search server* button. This will delete all the settings.

When you have completed all search server settings, click the *Next* button.

Summary of the Components

The last dialog box of the installation wizard shows you a summary of the components that will be installed.

If you install a new WCM system, “Base system” is displayed on top of the list. The base system comprises, among other things, the class libraries of Livelink WCM Server and the files and directories shared by the individual components.

To start the installation, click the *Finish* button.

After the installation, the Administration server is already running. To start the other servers, execute the respective start scripts, see section “Starting Servers” on page 152.

User-Defined Installation of the WCM System

The description of the user-defined installation of the WCM system is based on the description of a distributed system with firewall in chapter “Concepts” of the Livelink WCM Server Administrator Manual. The WCM system to be installed in this example is to consist of the following components and have the following structure:

- A master Admin server and a master Content server are located on a computer behind a firewall (i.e. in the intranet). A second computer hosts a proxy Content server running as a web application in an application server. This server is used for editing and quality assurance of the WCM objects by means of the Content client.
- Outside the firewall (i.e. outside the company network), there is a proxy Content server, on which the Production view of the website is published, and a proxy Admin server. The proxy Admin server is needed by the proxy Content server for loading the server configuration on startup, logging in users to the WCM system, and checking the license. The proxy servers outside the firewall use a separate data storage.

> **Note:** Providing a proxy Admin server requires the separate installation of a second WCM system outside the firewall.

To install this scenario, proceed as follows:

1. Install the master WCM system behind the firewall.
2. Install the proxy Content server behind the firewall
3. Install the proxy WCM system outside the firewall.
4. Register the proxy Admin server in the configuration of the master Admin server.

Step 1 – Install master WCM system behind the firewall

For installing such a scenario, you first configure the master Admin server and the master Content server on a computer behind the firewall, i.e. in the intranet.

1. Proceed as described in section “Installing a WCM System (Minimum System)” starting on page 68, and work through the dialog boxes to configure the master Admin server.
2. In the *Options for the server installation* dialog box, click the radio button *User-defined installation*.

   Note: You are not offered any tab for installing specific servers. With the exception of the Administration server, all servers are installed by means of the *Add server* button.

3. Click the *Add server* button to set up the master Content server for the WCM system behind the firewall.

   For detailed information on the individual server parameters, see section “Configuring Servers” on page 91.

4. Confirm the server settings by clicking the *Next* button.
5. If you want to set up Search servers, enter the required parameters in the following dialog box. In order to install Content Miner or Lucene Search servers, you require according licenses.

   For detailed information on installing Search servers, refer to section “Configuring Search Servers” on page 98.

   The final dialog box shows the components that will be installed.

6. To start the installation, click the *Finish* button.

Note: The installation process is aborted if the WCM administrator cannot be created successfully or if the Administration server cannot be started or reached.

Step 2 – Install the proxy Content server behind the firewall

Afterwards, install the proxy Content server on a second computer behind the firewall. Proceed as described in section “Adding New Servers” starting on page 110 and follow the dialog boxes.

Step 3 – Install proxy WCM system outside the firewall

1. Install a second WCM system and configure the second Administration server as a proxy server. In the dialog box with the data of the Administration server, deselect the *Master* check box, see section “Setting the Parameters for the New Administration Server” on page 74.

   For the proxy Admin server, too, you must configure a connection to a user administration. Make sure to enter a different pool name than for the master Admin server.

   Notes

   - The servers of the proxy system must have unique names that are not used in the master system.
   - Master and proxy Admin servers should access the same user administration, i.e. the same LDAP server, the same Livelink server, or the same database. For user administrations based on LDAP or an RDBMS, you can use separate servers provided that the user information is kept identical by means of synchronization between the servers. This synchronization does **not** belong to the tasks of the WCM system. The Admin servers must access the same type of user administration, e.g. both a database. You cannot mix different user administration types.
   - The initial administrator of the proxy WCM system must have **the same** user ID as the administrator of the master WCM system.

2. Here too, click the radio button *User-defined installation* and click the *Add server* button to set up the proxy Content server.

3. Confirm the server settings by clicking the *Next* button.
4. If you want to set up Search servers, enter the required parameters in the following dialog box. In order to install Content Miner or Lucene Search servers, you require according licenses.

   Note: When synchronizing the configuration of the proxy Admin server and the master Admin server, the settings of the proxy Admin server relating to the Search servers are overwritten. Before configuring the Search servers (e.g. adding collections), you must thus register the proxy Admin server in the configuration of the master Admin server (*Synchronize configuration* function in the Admin client).

   The final dialog box shows the components that will be installed.
5. To start the installation, click the *Finish* button.

Step 4 – Register the proxy Admin server in the configuration of the master Admin server

1. Start the Admin client on the master Admin server behind the firewall via the script **adminClient.bat** or **adminClient.sh**.
2. Register the proxy Admin server in the configuration of the master Admin server. For information on the exact procedure, refer to the Livelink WCM Server Administrator Manual.

   The Content server outside the firewall is now available in the master system. Make sure that the ports required for the communication between the master and proxy systems through the firewall are enabled.

Adding New Servers

You can add new Admin servers, Content servers, and Search servers to your WCM system at any time, for example if the infrastructure of your company changes.

Notes

- For installing additional servers, an installed and running Administration server is required, which must be accessible from the current computer.
- If you add servers to a WCM system managed by a proxy Admin server (proxy system), the master and proxy systems must be synchronized after the installation.

  No configuration changes must be made in the master system while new servers are being installed in the proxy system. Otherwise, the settings of the new servers in the proxy system would be overwritten when the configuration changes are automatically transferred by the master Admin server. To ensure that the configuration cannot be changed by other users, you can set the servers to run level 4 “Single user mode”

  For information on synchronizing the configuration, refer to the Livelink WCM Server Administrator Manual (chapter “Managing Servers”).
- If you set up two or more Content servers running in the JSP engine on the same computer, you must use different instances of your JSP engine for executing the servers. In this case, modify the default URL in the scripts for starting the respective Content servers according to the configuration of the JSP engine used after the installation of Livelink WCM Server (see section “Starting a Content Server in the Application Server” on page 159).

- Lucene Search servers can only be installed in the same directory as the assigned server.
- Please also note the information in section “Notes on installing” on page 68.

Procedure

New servers are added to an existing WCM system by means of the installation program. If you want to install the new server(s) on a different computer, the necessary installation files must be available on that computer.

To subsequently install new servers.

1. Start the installation program and select the directory for the installation.

   Note: If a Service Pack has already been installed for the WCM system, the command for starting the installation must be modified (see section“Calling the installation for a WCM system with Service Pack” on page 112).

2. In the *Type of installation* dialog box, click the *Add server* radio button (see section “Selecting the Type of Installation” on page 71) and click the *Next* button.
3. In the *Admin server* dialog box, enter the parameters of the responsible Admin server. This is necessary for establishing a connection to this server.

   You can check these parameters in the Admin client via *Configuration* → *Pools* → *WCM* → *{Admin server pool}*.
4. Click the *Check* button.
5. You must log in to the Admin server. Enter the user ID and the password of the responsible administrator.

6. Confirm the *Admin server* dialog box by clicking the *Next* button.
7. The next dialog box automatically offers you the option *User-defined installation*. Click the *Next* button.

   The *Server* dialog box opens. Here you can make settings for the new server.

   > **Note:** You are not offered any tabs for installing specific components. Install the new server by clicking the *Add server* button.

   For detailed information on the individual parameters, refer to section “Configuring Servers” on page 91.
8. To add another server, click the *Add server* button. This automatically opens a new tab on which you can enter the required parameters.

   If you want to remove the entries for a server, go to the appropriate tab and click the *Remove server* button.
9. Confirm the settings by clicking the *Next* button.

   The next steps correspond to the minimum installation of a WCM system.

The entries in the start script **startvip.bat** or **startvip.sh** are modified automatically after adding individual servers. Thus, the new server can be started by means of this script (see section “Starting Servers” on page 152).

Calling the installation for a WCM system with Service Pack

To extend a WCM system for which a Service Pack has already been installed, the command for starting the installation must be modified. Proceed as follows:

1. Copy the files **vipcore.jar**, **vipacs.jar**, and **vipapi.jar** from the directory **{WCM installation directory}\lib** to a local directory.

2. Start the installation with the following command:

   Windows

```
install.bat {SDK installation directory}\bin {path to the
defaults.xml file including file name} -INSTALLCLASSPATH {path to
   directory from step 1}
```

   UNIX

```
sh ./install.sh -jdk {SDK installation directory}/bin
   -defaults {path to the defaults.xml file including file name}
   -INSTALLCLASSPATH {path to the directory from step 1}
```

Installing the Admin Client

The installation program offers you the possibility to install the Admin client on a computer on which no WCM system is located.

Notes

- For installing an additional Admin client, an installed and running Administration server is required, which must be accessible from the current computer.
- Please also note the information in section “Notes on installing” on page 68.
- If a Service Pack has already been installed for the WCM system, the command for starting the installation must be modified (see section “Calling the installation for a WCM system with Service Pack” on page 112).

- You may also use the Admin client without installing it on your local computer. For this purpose, you create a web application for the Admin client and integrate this web application in an application server. For more information, refer to the Livelink WCM Server Administrator Manual.

Procedure

To install an additional Admin client:

1. Start the installation program and select the directory for the installation.
2. In the *Type of installation* dialog box, click the *Admin client* radio button (see section “Selecting the Type of Installation” on page 71) and click the *Next* button.
3. In the *Admin server* dialog box, enter the parameters of the responsible Admin server. This is necessary for establishing a connection to this server.

   You can check these parameters in the Admin client via *Configuration* → *Pools* → *WCM* → *{Admin server pool}*.
4. Click the *Check* button.
5. You must log in to the Admin server. Enter the user ID and the password of the responsible administrator.
6. Confirm the *Admin server* dialog box by clicking the *Next* button.
7. Click the *Finish* button.
8. To start the Admin client, call the file **adminClient.bat** or **adminClient.sh** in the directory **{WCM installation directory}\admin**.

Updating the License

Notes

- For updating the license, an installed and running Administration server is required, which must be accessible from the current computer.
- Please also note the information in section “Notes on installing” on page 68.
- If a Service Pack has already been installed for the WCM system, the command for starting the installation must be modified (see section “Calling the installation for a WCM system with Service Pack” on page 112).

Procedure

The license can be updated from any desired computer. To update the license:

1. Start the installation program and select the directory for the installation.
2. In the *Type of installation* dialog box, click the *Update license* radio button (see section “Selecting the Type of Installation” on page 71) and click the *Next* button.

   You are prompted to select the license file.
3. In the *License file* dialog box, you can specify the path to the new license file.

Notes:

You can modify the settings for the license expiration notification in the Admin client via *Configuration → Utilities → License expiration notification*.

You can check the status of your WCM licenses at any time via *System administration → Licenses*.

The installation program automatically checks whether the license file is valid.

4. Click the *Next* button.
5. In the *Admin server* dialog box, enter the parameters of the responsible Admin server. This is necessary for establishing a connection to this server.

   You can check these parameters in the Admin client via *Configuration → Pools → WCM → {Admin server pool}*.
6. Click the *Check* button.
7. You must log in to the Admin server. Enter the user ID and the password of the responsible administrator.
8. Confirm the *Admin server* dialog box by clicking the *Next* button.
9. Click the *Finish* button.

Directory Structure after the Installation

In the WCM installation directory, the installation program creates a number of directories for the different components of Livelink WCM Server.

Write permissions for the directories of Livelink WCM Server

Make sure that the processes of Livelink WCM Server have the necessary rights in the file system. The user who performs the installation of the WCM system needs full write rights for the WCM installation directory.

Also during operation, files are changed, created, and deleted. Thus, write access is required for the following subdirectories and files in the WCM installation directory:

- for the servers and the JSP engine: **\temp**, **\log**, **\website**, **\fet**
- for the Portal Manager API in connection with the Content client (Classic): **\data**
- for Search servers: **\contentminer**

  If the Search server is assigned to a Content server running in the JSP engine, the process of the JSP engine also needs write rights for this directory.
- for writing trace logs: **\trace**
- for the Admin server: **\config**
- for setting the one-time password in the stop scripts of the servers (see section “Stopping Servers” on page 157): **shutdown_{server name}.bat** or **.sh**

In the following, the most important directories and files in the WCM installation directory are briefly described.

Livelink WCM Server

The following directories and files are important:

- start scripts for the servers: **startvip.bat** (or **.sh**), **{server name}.bat** (or **.sh**), and **startserver.bat** (or **.sh**)
- script file for setting the class path: **setClasspath.bat** (or **.sh**). This file is called automatically in the context of the start scripts.
- scripts for stopping the servers: **shutdown_{server name}.bat** (or **.sh**)

> **Note:** For information on starting and stopping the servers and on the structure of the start scripts, refer to section “Starting and Stopping Servers” starting on page 151.

- Readme files for Livelink WCM Server

The installation directory contains a number of subdirectories:

- **\admin\:** contains the Admin client files and the files of the Export/Import tool
- **\config**: contains the configuration files of the WCM system

> **Important!** These files **must not** be changed manually. Otherwise, the WCM system can no longer be used.

- **\documentation**: contains the documentation on Livelink WCM Server in PDF format and the Javadoc on the programming interfaces of Livelink WCM Server
- **\examples**: contains examples, e.g. for programming server agents by means of the WCM Java API
- **\external_lib**: directory for external libraries which are integrated in Livelink WCM Server (e.g. JDBC drivers)

- **\fet**: contains important files for the deployment
- **\installation**: contains the log files created during installation. The subdirectory **\service** contains the files for setting up services for the servers.
- **\keys**: contains the server certificates of the WCM truststore
- **\latestpatch**: directory for the Service Pack files **vipacs.jar**, **vipapi.jar**, and **vipcore.jar**
- **\lib**: contains the class archives used by Livelink WCM Server
- **\log**: contains the log and error files of the servers (see section “Log Files and Error Files” on page 162) and of the Export/Import Tool
- **\tools**: contains the script file **precompile.bat** (or **.sh**) for precompiling the Content client (see section “The Precompile Script” on page 188). This directory also contains the script **setPomaClasspath.bat** (or **.sh**) for explicitly setting the class path. This script is required if the Content server is running in a JSP engine without its own class loader.
- **\VIPSecure**: contains the files for the access control component Secure Access
- **\website**: default directory for saving website data. The subdirectory **\cmsclient** contains the files of the Content client. The subdirectory **\htmlclient** contains the files of the Content client (Classic).

Search Servers

Content Miner Search servers

The following directories are important for Content Miner Search servers:

- **\contentminer\{Search server name}\server**: contains the start scripts for the Content Miner Search servers (see section “Starting Search Servers” on page 160)
- **\contentminer\{Search server name}\tools**: contains the preconfigured command line tools

Lucene Search servers

The following directories are important for Lucene Search servers:

- **\contentminer\{Search server name}\index**: contains the index
- **\contentminer\{Search server name}\tools**: contains the preconfigured command line tools

Livelink Search servers

The following directory is important for Livelink Search servers:

\livelinksearch\{Search server name}\tools: contains the preconfigured command line tools

Examples for using Search servers

The directory **\examples\SearchServer** contains simple examples for using Search servers. To be able to use these files, you must make the necessary modifications to the HTTP server and the JSP engine.

You can also import the files into an existing website by means of the **multiimport.zip** file supplied in the directory **\examples**. The start file for the import is **index.htm**. For information on performing imports from ZIP files, refer to the Content Client User Manual.

Tag Libraries

To enhance structuring of JSP pages, the JSP specification (version 1.1) provides so-called *tag libraries* (taglibs for short).

Some tag libraries are supplied with Livelink WCM Server. They are located in the directory **{WCM installation directory}\examples\taglets**. For information on programming examples, refer to the online documentation supplied (**ReadMe.html**). For further information on taglets, refer to the Content Miner Manual.

Deinstalling the WCM System

You can use the installation program to deinstall a complete WCM system or individual components of the system.

During a successful deinstallation, the installation program removes all files, directories, and registry entries automatically.

The installation program does, however, not delete the database tables created and used by the WCM components. These must be deleted manually after the deinstallation of the complete WCM system (see section “Deleting Database Tables” on page 125).

There are two possible ways of deinstalling components:

- by means of the WCM installation program, see the following section
- by means of the console, see section “Installation and Deinstallation via the Console” on page 126

Notes on the deinstallation

- Deinstalling a WCM system requires an installed and running Admin server, which must be accessible from the current computer.
- The directory in which you want to perform the deinstallation must contain components of a WCM system. Otherwise the option *Deinstall* is not activated.
- A server can only be deinstalled if there are no websites assigned to it any longer. If necessary, use the Admin client to delete the relevant websites before deinstalling the server, or assign the websites to a different server.

  If this is no longer possible because the respective server is damaged or cannot be reached for other reasons, the server cannot be removed via the installation program. In this case, use the function *Delete server* in the *Configuration* view of the Admin client (see Livelink WCM Server Administrator Manual).
- If a server that you want to remove has been assigned a Search server that is located in the same directory as the server, then that Search server will automatically be deinstalled as well. If the Search server is in a different directory than the server, it must be removed separately using the deinstallation procedure.
- If you remove individual servers, the information in the start script **startvip.bat** or **startvip.sh** is modified automatically.

Deinstalling Components via the Graphical User Interface

To deinstall components via the graphical user interface:

1. Start the installation program and select the installation directory.
2. In the *Type of installation* dialog box, click the *Deinstall* radio button (see section “Selecting the Type of Installation” on page 71) and click the *Next* button.
3. In the *Admin server* dialog box, enter the parameters of the responsible Admin server. This is necessary for establishing a connection to this server.

   You can check these parameters in the Admin client via *Configuration* → *Pools* → *WCM* → *{Admin server pool}*.
4. Click the *Check* button.
5. You must log in to the Admin server. Enter the user ID and the password of the responsible administrator.
6. Confirm the *Admin server* dialog box by clicking the *Next* button.

   The next dialog box shows you the components of your WCM system that can be deinstalled. If the servers that you want to remove are displayed as inactive, there may still be websites assigned to them.

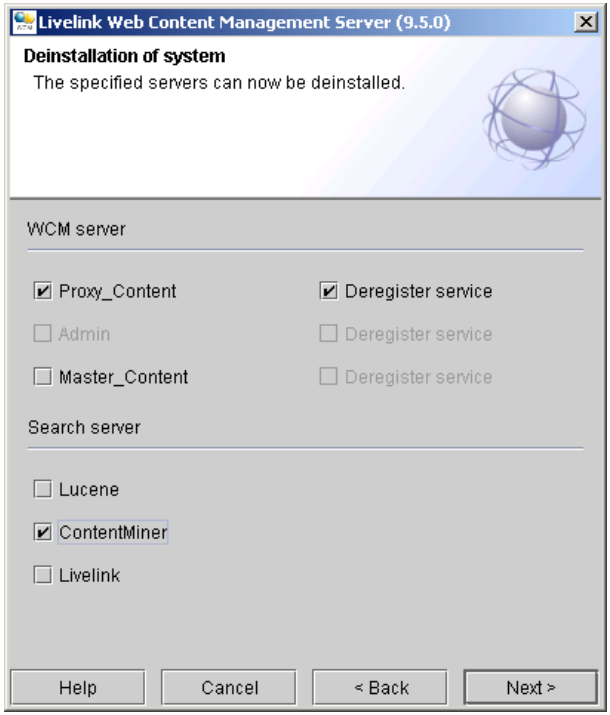


Fig. 19 – Selecting the components to be removed

7. Select the check boxes for the components that are to be removed from the WCM system.

   For servers, you can additionally select the *Deregister service* check box. This removes the entry for automatically starting the server from the Windows registry.

   > **Note:** By means of the **uninstall_{server name}.bat** scripts in the directory **{WCM installation directory}\installation\service**, you can remove the services from the Windows registry independent of the server deinstallation.

8. Click the *Next* button.

The last dialog box of the installation wizard shows you a summary of the components that will be deinstalled.

9. Click the *Finish* button.

Deleting Database Tables

Once you have completely removed the WCM system, the tables used by Livelink WCM Server must be removed from the database. The following description refers to the RDBMS Oracle.

Delete the following tables in the Oracle database. You can use the SQL command `drop table {name}` for this purpose.

- ADMIN_FIELDINFO
- ADMIN_FUNCAREA_GROUP
- ADMIN_FUNCAREA_ROLE
- ADMIN_FUNCAREA_USER
- ADMIN_GROUP
- ADMIN_HCL
- ADMIN_ROLE
- ADMIN_SUBST_USER
- ADMIN_USER
- ADMIN_USER_GROUP
- ADMIN_USER_ROLE
- ADMIN_VIP_VER
- ADMIN_WEBSITE_GROUP
- ADMIN_WEBSITE_ROLE
- ADMIN_WEBSITE_USER

Installation and Deinstallation via the Console

As an alternative to the installation via the graphical user interface, you can also install and deinstall the WCM system and individual components via the console.

> **Note:** During the installation via the console, the installation program does not make as many checks for the correctness of the entries in the file **defaults.xml** as during the installation via the graphical user interface. This gives you more choices during the installation. However, it may also lead to problems.

Parameters in the File `defaults.xml`

By default, the parameters required for installing and deinstalling are read from the file **defaults.xml**, which is located in the directory **\installation** on the WCM CD. This file contains all default settings that are used during installation and deinstallation. Copy the file to a local drive and adapt the entries to your requirements.

The **defaults.xml** file is subdivided into several sections for the different components of Livelink WCM Server. By means of the parameter `<install_action>`, you can specify for each component whether it is to be installed, deinstalled, or not to be changed by the installation program.

> **Note:** The parameter `<install_action>` in the section `<common>` controls the installation option for the base system. Enter `install` if the entire WCM system or individual components are to be installed. Only enter `uninstall` if the **entire** WCM system is to be removed.

Different scripts are provided for installation and deinstallation (see section “Starting the Installation via the Console” on page 149 and section “Starting the Deinstallation via the Console” on page 150.

The following example illustrates the settings required for installing a master Admin server with RDBMS data storage and a master Content server running as a web application in an application server. This scenario corresponds to a minimum installation.

Note: The installation process is aborted if the WCM administrator cannot be created successfully or if the Admin server cannot be started or reached.

```
<default>

      <license>
         <directory>license.xml</directory>
         <mailto>wcmadmin@company.example</mailto>
         <daysbefore>10</daysbefore>
         <percentage>80</percentage>
      </license>

      <common>
         <language>en_US</language>
         <directory>d:\wcm</directory>
         <directory_unix>/export/home/wcmuser/wcmsystem/wcm
            </directory_unix>
         <vipadminuser>admin</vipadminuser>
         <vipadminpassword>admin</vipadminpassword>
         <default_suffix>htm</default_suffix>
         <server_languages>en_US de_DE</server_languages>
         <install_action>install</install_action>
      </common>

      <admin>
         <name>Admin</name>
         <server_category>master</server_category>
         <host>wcmserver.company.example</host>
         <http_port>5003</http_port>
         <vipp_port>5002</vipp_port>
         <secure>false</secure
         <storage>rdbms</storage>
         <mailhost>mailserver.company.example</mailhost>
         <mailsender>wcm@company.example</mailsender>
         <nt_service>false</nt_service>
         <install_action>install</install_action>
      </admin>
```

```
    <server name="Master_Content">
       <server_type>content</server_type>
       <server_category>master</server_category>
       <host>wcmserver.company.example</host>
       <http_port>5005</http_port>
       <vipp_port>5004</vipp_port>
       <secure>false</secure>
       <encoding>UTF-8</encoding>
       <nt_service>false</nt_service>
       <create_application_file>true</create_application_file>
       <application_file_dir>d:\temp</application_file_dir>
       <application_name>wcm</application_name>
       <application_with_client>true</application_with_client>
       <application_client_name>cmsclient</application_client_name>
       <application_file_template>web23.xml
          </application_file_template>
       <application_servlet></application_servlet>
       <application_taglet></application_taglet>
       <application_filter></application_filter>
       <application_encoding>UTF-8</application_encoding>
       <application_default>false</application_default>
       <install_action>install</install_action>
    </server>

    <rdbms>
       <poolname>userpool</poolname>
       <user>wcmuser</user>
       <password>wcm</password>
       <type>ORACLE</type>
       <db_owner></db_owner>
       <jdbc_driver>oracle.jdbc.driver.OracleDriver</jdbc_driver>
       <driverclasspath></driverclasspath>
       <connection_string>
          jdbc:oracle:thin:@dbserver.company.example:1521:wcmdb
       </connection_string>
       <check>true</check>
    </rdbms>

</default>
```

The file **defaults.xml** is subdivided into several main sections:

- `<license> </license>`, see section “Entries in the `<license>` Section” on page 130

- <common> </common>, see section “Entries in the <common> Section” on page 131
- <admin> </admin>, see section “Entries in the <admin> Section” on page 132
- <update_vip> </update_vip>, see section “Entries in the <update_vip> section” on page 222
- <update_rdbms> </update_rdbms>, see section “Entries in the <update_rdbms> section” on page 224
- <server> </server>, see section “Entries in the <server> Section” on page 134
- <contentminer_server> </contentminer_server>, see section “Entries in the <contentminer_server> Section” on page 138
- <lucene_server> </lucene_server>, see section “Entries in the <lucene_server> Section” on page 140
- <livelink_search-server> </livelink_search-server>, see section “Entries in the <livelink_search-server> Section” on page 141
- <external_libs> </external_libs>, see section “Entries in the <external_libs> Section” on page 142
- <rdbms> </rdbms>, see section “Entries in the <rdbms> Section” on page 142
- <livelink> </livelink>, see section “Entries in the <livelink> Section” on page 144
- <ldap> </ldap>, see section “Entries in the <ldap> Section” on page 146
- <converter> </converter>: information for the LDAP directory service, e.g. on mapping WCM attributes to LDAP attributes (see section“Mapping WCM Attributes to LDAP Attributes” on page 42)

The following sections explain the individual entries.

Entries in the <license> Section

This section contains information on the license file and some parameters for license management.

> **Note:** Please also note the information on these parameters in section “Specifying the License Management Options” on page 72.

Table 9 – Entries in the <license> section

Entry in the XML file	Explanation
<directory>license.xml</directory>	Complete path to the license file
<mailto>wcmadmin@company.example</mailto>	E-mail address of the person who is to be notified if the WCM licenses expire or are exceeded
<daysbefore>10</daysbefore>	Option for sending the license expiration notification: number of days before the licenses expire
<percentage>80</percentage>	Option for sending the license expiration notification: percentage of licenses used by active users

Entries in the <common> Section

This section contains general installation parameters, such as language and directory, information on the WCM administrator, and the installation option for the base system.

Table 10 – Entries in the <common> section

Entry in the XML file	Explanation
`<language>en_US</language>`	The language of the installation dialog boxes and the initial language of the administrator Possible values: language settings in the format `language code_country code` according to ISO-639 and ISO-3166
`<directory>d:\wcm</directory>`	Target directory under Windows the WCM system is to be installed in
`<directory_unix>/home/wcmuser/wcmsystem/wcm</directory_unix>`	Target directory under UNIX the WCM system is to be installed in
`<vipadminuser>admin</vipadminuser>`	User ID of the WCM administrator See section “Setting up the WCM Administrator” on page 88
`<vipadminpassword>admin</vipadminpassword>`	Password of the WCM administrator
`<default_suffix>htm</default_suffix>`	Default file extension for generated pages See section “Setting the Parameters for the New Administration Server” on page 74

Entry in the XML file	Explanation
`<server_languages>en_US de_DE</server_languages>`	The base language(s) of the servers Possible values: language settings in the format `language code_country code` according to ISO-639 and ISO-3166
`<install_action>install</install_action>`	Installation option for the base system Possible values: `install` (when adding components to the WCM system), `uninstall` (only when deinstalling the entire WCM system), `none` (no change, e.g. when removing individual components from an existing WCM system or when upgrading the WCM system) For information on the base system, also refer to section “Summary of the Components” on page 105.

Entries in the <admin> Section

This section contains the settings of the Administration server.

> **Note:** Please also note the information on the individual parameters in section “Setting the Parameters for the New Administration Server” on page 74.

Table 11 – Entries in the <admin> section

Entry in the XML file	Explanation
`<name>Admin</name>`	Name of the Administration server

Entry in the XML file	Explanation
`<server_category>master</server_category>`	Server category of the Administration server Possible values: `master` and `proxy`
`<host>wcmserver.company.example</host>`	Fully qualified host name of the computer on which the Administration server is to be installed
`<http_port>5003</http_port>`	Port of the Admin server for the HTTP connection (HTTP tunneling)
`<vipp_port>5002</vipp_port>`	Port of the Admin server for the VIPP connection
`<secure>false</secure>`	Indicates whether communication between the servers is to take place via secure connections (Secure Socket Layer) Possible values: `true` or `false`
`<storage>rdbms</storage>`	Storage method for the user data Possible values: `rdbms`, `ldap`, or `livelink`
`<mailhost>mailserver.company.example</mailhost>`	Fully qualified host name of the SMTP server
`<mailsender>wcm@company.example</mailsender>`	Mail address used as sender address for e-mails sent by the system

Entry in the XML file	Explanation
`<nt_service>false</nt_service>`	Indicates whether the Administration server is to be set up as a Windows service. This service can be used for automatic server startup and shutdown under Windows. Possible values: `true` or `false`
`<install_action>install</install_action>`	Installation option for the Administration server Possible values: `install` (installation), `uninstall` (deinstallation), `none` (no change, e.g. when adding servers to an existing WCM system or when upgrading the WCM system) If you enter `none`, the installation program automatically checks the connection to the master Admin server.

Entries in the `<server>` Section

This section contains information on the Content servers to be installed. For each Content server to be installed, you must fill in a separate paragraph.

> **Note:** Please also note the information on these parameters in section “Configuring Servers” on page 91.

Table 12 – Entries in the `<server>` section

Entry in the XML file	Explanation
`<server name="Master_Content">`	Name of the Content server
`<server_type>content</server_type>`	Server type The value is always `content`.
`<server_category>master</server_category>`	Server category Possible values: `master` and `proxy`
`<host>wcmserver.company.example</host>`	Fully qualified host name of the computer on which the Content server is to be installed
`<http_port>5005</http_port>`	Port of the Content server for the HTTP connection (HTTP tunneling)
`<vipp_port>5004</vipp_port>`	Port of the Content server for the VIPP connection
`<secure>false</secure>`	Indicates whether communication between the servers is to take place via secure connections (Secure Socket Layer) Possible values: `true` or `false`

Entry in the XML file	Explanation
`<encoding>UTF-8</encoding>`	Only available for Content servers running in the JSP engine Encoding for the Content client Possible values: `UTF-8` (Unicode), `ISO-8859-1` (Latin-1)
`<nt_service>false</nt_service>`	Not available for Content servers running in the JSP engine Indicates whether a Windows service is to be set up for this server This service can be used for automatic server startup and shutdown under Windows. Possible values: `true` or `false`
`<create_application_file>false</create_application_file>`	Indicates whether a web application is to be generated for this server. For working with the Content client or the Content client (Classic), at least one Content server must run in the context of a JSP engine or as a web application in an application server. Possible values: `true` or `false`
`<application_file_dir>d:\temp<application_file_dir>`	Directory for saving the generated WAR file
`<application_name>wcm</application_name>`	Name of the generated web application. This name determines the name of the WAR file and becomes part of the URL used to call the web application in the application server.
`<application_with_client>true</application_with_client>`	Indicates whether the Content client is to be integrated in the web application Possible values: `true` or `false`

Entry in the XML file	Explanation
`<application_client_name>cmsclient</application_client_name>`	Only available if the entry `<application_with_client>` has the value `true` Name for the Content client. Like the name of the web application, the name of the Content client is added to the URL for calling the Content client.

`<application_file_template>web23.xml</application_file_template>`	Application server used. Instead of a product, a servlet API standard may be entered.
	
	Possible values:
	
	`servlet-api22.xml`
	`servlet-api23.xml`
	`websphere4.xml`
	`websphere5.xml`
	`weblogic81.xml`
	`tomcat4110.xml`
	`oracle9i.xml`
	`resin214.xml`
	`jrun4.xml`
	`servletexec42.xml`

`<application_servlet></application_servlet>`
`<application_taglet></application_taglet>`
`<application_filter></application_filter>`
	Mapping of servlets, tag libraries, or filter servlets that are not used in Livelink WCM Server, but are to be available in the generated web application
	
	Note: In these parameters, valid XML must be entered and the characters `<` and `>` must be encoded as `<` and `>` respectively (example: `<servlet>` becomes `<servlet>`).

Entry in the XML file	Explanation
`<application_encoding>UTF-8</application_encoding>`	Encoding for the web application Possible values: `UTF-8` (Unicode), `ISO-8859-1` (Latin-1)
`<application_default>false</application_default>`	Indicates whether the generated web application is defined as default application in the application server. The default application is addressed via the root URL of the application server, i.e. the URL does not contain the application name. Possible values: `true` or `false`
`<install_action>install</install_action>`	Installation option for the server Possible values: `install` (installation), `uninstall` (deinstallation), `none` (no change, e.g. when upgrading the WCM system)

Entries in the `<contentminer_server>` Section

This section contains information on the Content Miner Search servers.

Note: Please also note the information on these parameters in section “Tab Content Miner” on page 98.

Table 13 – Entries in the `<contentminer_server>` section

Entry in the XML file	Explanation
`<contentminer_server name="Contentminer1">`	Name of the Content Miner Search server
`<host>wcmserver.company.example</host>`	Fully qualified host name of the computer on which the Content Miner Search server is to be installed
`<query>9000</query>`	Connection port of the Query server
`<index>9001</index>`	Connection port of the Index server
`<vip_server>Master-Content</vip_server>`	Name of a Content server that already exists or is to be installed and to which you want to assign this Search server
`<install_action>install</install_action>`	Installation option for the Search server Possible values: `install` (installation), `uninstall` (deinstallation), `none` (no change, e.g. when upgrading the WCM system)

Entries in the <lucene_server> Section

This section contains information on the Lucene Search servers.

> **Note:** Please also note the information on these parameters in section “Lucene tab” on page 101.

Table 14 – Entries in the <lucene_server> section

Entry in the XML file	Explanation
`<lucene_server name="Lucene1">`	Name of the Lucene Search server
`<reload_interval>300000</reload_interval>`	Interval in milliseconds after which all collections will be reloaded for read access
`<vip_server>Master-Content</vip_server>`	Name of a Content server that already exists or is to be installed and to which you want to assign this Search server

`<index_location>`
`<server name="Master_Content">d:\temp</server>`
`</index_location>`
	Path to the directory in the file system in which the index is saved. This storage location must be available for all servers. The storage location is set for each server individually. The paths must point to the same directory for all servers.

`<install_action>install</install_action>`	Installation option for the Search server Possible values: `install` (installation), `uninstall` (deinstallation), `none` (no change, e.g. when upgrading the WCM system)

Entries in the <livelink_search-server> Section

This section contains information on the Livelink Search servers.

> **Note:** Please also note the information on these parameters in section “Tab Livelink” on page 103.

Table 15 – Entries in the <livelink_search-server> section

Entry in the XML file	Explanation
`<livelink_search-server name="Livelink1">`	Name of the Livelink Search server
`<vip_server>Master-Content</vip_server>`	Name of a Content server that already exists or is to be installed and to which you want to assign this Search server

`<url>`
`http://livelink.company.example/livelink/livelink.exe`
`</url>`
	URL to the Livelink server which is to perform the indexing and to process the search requests

`<host>livelink.company.example</host>`	Fully qualified name of the computer hosting the Livelink server which is to perform the indexing and to process the search requests
`<port>2099</port>`	Connection port on the Livelink server
`<urlusable>true</urlusable>`	Indicates whether to use the URL of the Livelink server for the connection. The entries <host> and <port> will be ignored in this case.

Entry in the XML file	Explanation
`<username>Admin</username>`	User for processing search requests in the Livelink system. The user must have read access to the slices (collections).
`<password>admin</password>`	Password for processing search requests in the Livelink system
`<install_action>install</install_action>`	Installation option for the Search server Possible values: `install` (installation), `uninstall` (deinstallation), `none` (no change, e.g. when upgrading the WCM system)

Entries in the <external_libs> Section

In this section, you can specify the path to the external libraries which the installation program is to copy to the directory **{WCM installation directory}\external_lib**. Use semicolons (Windows) or colons (UNIX) to separate multiple paths.

Entries in the <rdbms> Section

This section contains the parameters for the connection from the Administration server to the relational database management system. For each RDBMS supported, the **defaults.xml** file contains a separate paragraph with default settings.

> **Note:** Please also note the information on these parameters in section “Setting RDBMS Parameters” on page 76.

Table 16 – Entries in the <rdbms> section

Entry in the XML file	Explanation
`<poolname>userpool</poolname>`	Name of the JDBC pool created for the connections to the RDBMS
`<user>wcmuser</user>`	User ID used by the WCM system for logging in to the database
`<password>wcm</password>`	Password used by the WCM system for logging in to the database
`<type>ORACLE</type>`	Type of the database management system Possible values: `ORACLE`, `MS SQLServer2000`
`<db_owner></db_owner>`	Only for SQL Server Owner of the database, i.e. ID of the user who created the respective tables
`<jdbc_driver>oracle.jdbc.driver.OracleDriver</jdbc_driver>`	Name of the driver class for the RDBMS used
`<driverclasspath></driverclasspath>`	Absolute path to the file(s) with the JDBC drivers for the database used. If the driver consists of several files, enter the paths separated by semicolons (;). The installation program copies the file(s) with the JDBC drivers to the directory **{WCM installation directory}\external_lib**. This is done on all servers. Example: `msbase.jar;mssqlserver.jar;msutil.jar` (Microsoft driver for MS SQL Server)

Entry in the XML file	Explanation

`<connection_string>`
`jdbc:oracle:thin:@dbserver.company.example:1521:wcmdb`
`</connection_string>`
	Full name of the data source. This parameter contains the fully qualified host name of the database server, the port for the connection to the database server, and the name of the database.

`<check>true</check>`	Specifies whether the connection between the master Admin server and the database management system is to be checked. When installing a proxy Admin server, the parameter must be set to `false`. Possible values: `true` or `false`

Entries in the <livelink> Section

This section contains the parameters for the connection from the Administration server to the Livelink user administration.

Note: Please also note the information on these parameters in section “Specifying the Parameters of the Livelink System” on page 80.

Table 17 – Entries in the <livelink> section

Entry in the XML file	Explanation
`<poolname>userpool</poolname>`	Name of the Livelink pool created for the connections to the Livelink system

Entry in the XML file	Explanation
`<host>livelink.company.example</host>`	Fully qualified name of the computer hosting the Livelink server
`<port>2099</port>`	Port number for the connection to the Livelink server This value corresponds to the value of the variable `Port` in the `[general]` section of the **opentext.ini** file (default value: 2099).
`<user>Admin</user>`	User ID for access to the Livelink server. The user must be allowed to add, modify, and delete users and groups in the Livelink system.
`<password>admin</password>`	Password for access to the Livelink server
`<database></database>`	Logical name of the used Livelink database. Normally, this field remains empty, i.e. the database configured as default database in Livelink will be used.

`<secure>false</secure>`
`<https>true</https>`
`<livelinkcgi>/intranet/llisapi.dll</livelinkcgi>`
`<httpusername>Admin</httpusername>`
`<httppassword>password</httppassword>`
`<verifyserver>true</verifyserver>`
`<domainname />`
	Important! These entries must not be changed.

Entry in the XML file	Explanation
`<check>false</check>`	Specifies whether the connection between the master Admin server and the Livelink server is to be checked Possible values: `true` or `false`

Entries in the <ldap> Section

This section contains the parameters for the connection of the Administration server to the LDAP directory service.

Notes:

Please also note the information on these parameters in sections “Setting the Parameters for the LDAP Directory Service” on page 82 and “Specifying WCM-Specific LDAP Parameters” on page 85.

The mapping of WCM attributes to LDAP attributes is also specified in the **defaults.xml** file, see section “Mapping WCM Attributes to LDAP Attributes” on page 42.

Table 18 – Entries in the <ldap> section

Entry in the XML file	Explanation
`<poolname>userpool</poolname>`	Name of the LDAP pool created for the connections to the LDAP directory service
`<provider>com.sun.jndi.ldap.LdapCtxFactory</provider>`	Driver for the connection to the LDAP directory service Possible value: `com.sun.jndi.ldap.LdapCtxFactory`

Entry in the XML file	Explanation
`<searchbase>ou=software solutions, o=company.example</searchbase>`	Search node. This parameter can be used to limit the search to certain branches of the LDAP directory service.
`<searchlimit>1000</searchlimit>`	Maximum number of results Livelink WCM Server retrieves for a directory search
`<authentication>simple</authentication>`	Authentication procedure used by the LDAP directory service Possible value: `simple`
`<url>ldap://LDAPserver.company.example:389</url>`	Address for access to the LDAP server
`<user>cn=admin, o=company.example</user>`	Distinguished name of the user for the LDAP binding profile used by the WCM system for logging in to the LDAP directory service
`<password>adminadmin</password>`	Password of the LDAP binding profile
`<userclass>vipUser</userclass`	LDAP object class for storing user data Default value: `vipUser`
`<groupclass>vipGroup</groupclass>`	LDAP object class for storing group data Default value: `vipGroup`

Entry in the XML file	Explanation
`<roleclass>vipRole</roleclass>`	LDAP object class for storing role data Default value: `vipRole`
`<nestedgroups>false</nestedgroups>`	Indicates whether the implicit assignment of users to groups and roles is to be considered in the WCM system (see section “Concept of Collective Groups and Collective Roles” on page 36) Possible values: `true` or `false`
`<secure>false</secure>`	Specifies whether communication with the LDAP server is to be performed via secure connections (Secure Socket Layer) Possible values: `true` or `false`
`<servertype>other</servertype>`	Indicates whether Microsoft Active Directory is used Possible values: `ads` or `other`
`<check>true</check>`	Specifies whether the connection between the master Admin server and the LDAP server is to be checked Possible values: `true` or `false`

Starting the Installation via the Console

> **Note:** Please note the information in section “Notes on installing” on page 68.

- **Windows:** Open the MS-DOS prompt and switch to the CD-ROM drive or to the directory with the installation files. Call the script with the path to your Java 2 SDK directory and the path to the **defaults.xml** file (including file name):

```
installConsole.bat {SDK installation directory}\bin
   {path to defaults.xml file}
```

- **UNIX:** The root directory of the WCM CD contains a JAR archive. This archive must be copied to your server and extracted to a temporary directory. Change to this directory and call the installation script with the path to your Java 2 SDK directory and the path to the **defaults.xml** file (including file name) as parameters.

```
sh ./installConsole.sh [-jdk {SDK installation directory}/bin]
   [-defaults {path to defaults.xml file}]
```

  If you call the installation script without any parameters, you will be asked to enter the path to your Java 2 SDK directory (**{SDK installation directory}/bin/**) and to the **defaults.xml** file.

The parameters for the installation are read from the **defaults.xml** file. For all components to be installed, the value `install` must be entered in the `<install_option>` tag in the respective sections. For components that are neither to be installed nor deinstalled, `none` must be entered in this tag (see section “Parameters in the File `defaults.xml`” starting on page 126).

The results of the installation are written to the installation log, see section “Log Files and Error Files” on page 162.

Starting the Deinstallation via the Console

> **Note:** Please note the information in section “Notes on the deinstallation” on page 122.

- **Windows:** Open the MS-DOS prompt and switch to the CD-ROM drive or to the directory with the installation files. Call the script with the path to your Java 2 SDK directory and the path to the **defaults.xml** file (including file name):

```
uninstallConsole.bat {SDK installation directory}\bin
   {path to defaults.xml file}
```

- **UNIX:** The root directory of the WCM CD contains a JAR archive. This archive must be copied to your server and extracted to a temporary directory. Change to this directory and call the uninstallation script with the path to your Java 2 SDK directory and the path to the **defaults.xml** file (including file name) as parameters.

```
sh ./uninstallConsole.sh -[jdk {SDK installation directory}/bin]
   [-defaults {path to defaults.xml file}]
```

  If you call the deinstallation script without any parameters, you will be asked to enter the path to your Java 2 SDK directory (**{SDK installation directory}/bin/**) and to the **defaults.xml** file.

The parameters for the deinstallation are read from the file **defaults.xml**. For all components to be deinstalled, the value `uninstall` must be entered in the `<install_option>` tag in the respective paragraphs. For components that are neither to be installed nor deinstalled, `none` must be entered in this tag (see section “Parameters in the File `defaults.xml`” starting on page 126).

The results of the deinstallation are written to the log, see section “Log Files and Error Files” on page 162.

Starting and Stopping Servers

The servers of the WCM system only need to be started once and should then remain permanently in operation. It is advisable to run them as background processes. It is important that the user under whose name the processes are started possesses the necessary write and read permissions in the relevant WCM directories (see section “Write permissions for the directories of Livelink WCM Server” on page 117.

During installation of the WCM system as well as during subsequent installation of Admin servers, Content servers, and Search servers, scripts for starting and stopping the servers are created. The various servers are started and stopped differently:

- starting Admin and Content servers (see sections “Starting Servers” on page 152 and “Stopping Servers” on page 157)
- starting a Content server running in the context of a JSP engine or as a web application in an application server (see section “Starting a Content Server in the Application Server” on page 159)
- starting Search servers (see section “Starting Search Servers” on page 160)

Important! The supplied start and stop scripts for UNIX comply with the standard POSIX/IEEE Std 1003.1-2001 (see http://www.opengroup.org/onlinepubs/007904975/toc.htm). If necessary, the scripts must be adapted to the UNIX variant used, the version of the operating system, and the shell.

Starting Servers

Important! The servers may only be started under the user name under which they were installed. They should **not** be started under the user ID of the administrator. Under UNIX, you should not operate the servers with the uid “0” or “root”.

First start the Admin server via **{Admin server name}.bat** or **{Admin server name}.sh**. Directly after the installation, the Admin server is already running. Afterwards, the servers can be started together via one start script or via separate start scripts.

The following notes apply to Admin and Content servers. For special notes on Content servers running in the context of a JSP engine or as a web application in an application server and on Search servers, refer to sections “Starting a Content Server in the Application Server” on page 159 and “Starting Search Servers” on page 160.

Starting Servers Together

For starting all servers (except for the Admin server which must already be running), the script **startvip.bat** or **startvip.sh** is provided.

- **Windows:** Open (e.g. in Windows Explorer) the WCM installation directory and run the script **startvip.bat**.
- **UNIX:** Switch to the WCM installation directory and run the script:

```
sh ./startvip.sh
```

Under UNIX, you can also label the server start files as executable programs. For this purpose, the file **set_executable.sh** located in the WCM installation directory is used. Run this file before starting the HTTP server, JSP engine, and Admin server.

In this case, the start script is called as follows:

```
./startvip.sh
```

Note: If you operate the servers under UNIX, please make sure that the number of file descriptors available to the Java process amounts to at least 1024.

Starting Servers Separately

The individual servers can be started separately in the following ways:

- via the server start script **{server name}.bat** or **{server name}.sh** of the server
- via the script **startserver.bat** or **startserver.sh**

Starting via server start scripts

To start individual servers, execute the respective script on the console. The script is named after the server – if the master Content server is called “Master_Content”, the name of the associated script is **Master_Content.bat** or **Master_Content.sh**.

Example – Start script of a master Content server (the default ports of the Admin server are used):

```
d:
cd "d:\wcm"

call "d:\wcm\setClasspath.bat"

start "Master_Content" java -Xmx512m -Dvip.class.path=%VIPCLASSPATH%
  de.gauss.vip.contentmanager.server.ServerStarter
    -name Master_Content -admin {Admin host} -vipp 5002 -http 5003
      -language en_US -secure false

exit
```

The start scripts are created automatically during installation of the servers. The following table gives an overview of the required and possible parameters in the start scripts of the servers.

Table 19 – Parameters in the server start scripts

Parameter	Explanation	Optional
`-Xmx512m`	Specifies the maximum memory size in megabytes which the server may take up. **Note:** When installing two or more servers on a computer, make sure that the total value for all installed servers does not exceed the size of the virtual memory.	

Parameter	Explanation	Optional
`-Dvip.vipsecure.debug=true`	Must be entered in front of the class name `de.gauss.vip.contentmanager.server.ServerStarter` in the start file If this parameter is specified, the requests and responses between this server and Secure Access are saved in the log file **{server name}_out.log**.	✓
`-name`	Name of the server	
`-admin`	(not available for Administration servers) Name of the computer hosting the responsible Admin server	
`-vipp`	(not available for Administration servers) VIPP port of the Admin server	
`-http`	(not available for Administration servers) HTTP port of the Admin server	
`-language`	Default language of the server Value: `de_DE` or `en_US`	
`-secure`	Use of SSL `true` – SSL is used `false` – SSL is not used	
`-?` `-help`	Prints out call parameters (help text) and exits script	✓

Parameter	Explanation	Optional
`-console`	Permits to enter commands via the console (debugging)	✓
`-installdir`	Installation directory	
`-logsize`	Maximum size of log files for Livelink WCM Server in KB Standard = 10240, minimum = 5	✓
`-maxlogs`	Number of log files to be kept in store Standard = 3, minimum = 0	✓
`-noredirect`	Messages are not recorded in the log files, but output to the console.	✓

Starting by means of the script startserver.bat (or startserver.sh)

Alternately, you can start individual servers via the script **startserver.bat** or **.sh**.

The script **startserver** (with default ports of the Admin server):

```
d:
cd "d:\wcm"

call "d:\wcm\setClasspath.bat"

SET SERVERNAME=%1%

if "%SERVERNAME%" == "" goto ERROR

start "%SERVERNAME%" java -Xmx256m -Dvip.class.path=%VIPCLASSPATH%
    de.gauss.vip.contentmanager.server.ServerStarter -name %SERVERNAME%
      -admin {Admin host} -vipp 5002 -http 5003 -language en_US
          -secure false

exit
```

```
:ERROR
echo improper call to this batch.
echo usage: "startserver <servername>"
```

The individual parameters in the script correspond to the entries in the server start scripts, see table “Parameters in the server start scripts” on page 154.

To start a server by means of this script:

1. Open the console.
2. Change to the WCM installation directory.
3. Enter:

   ```
   startserver {server name}
   ```

   for example:

   ```
   startserver Admin
   ```

Stopping Servers

For shutting down the servers, the scripts **shutdown_{server name}.bat** or **shutdown_{server name}.sh** are provided. Alternately, you can shut down the servers via the Admin client.

Table 20 – Parameters in the server stop scripts

Parameter	Explanation	Optional
-HOSTNAME	Fully qualified name of the computer hosting the server	
-vipp	VIPP port of the server	
-HTTP	HTTP port of the server	

Parameter	Explanation	Optional
`-SECURE`	Use of SSL `true` – SSL is used `false` – SSL is not used	
`-DATA`	Complete path to the file containing the encrypted one-time password	
`-runlevel`	Run level to which the server is to be set. It is also possible to restart the server by means of this parameter. The default value is 0 = “Server not available”.	✓
`-website`	Website to which the value set in the `-runlevel` parameter refers to The default value is “all websites”.	✓

Notes

- The stop scripts are protected against manipulation by means of a one-time password.
- When you shut down an Administration server by means of the script, the other servers will not shut down. However, without a running Administration server, it is not possible to log in to the WCM system.
- Run level changes in the master system do not affect WCM systems managed by a proxy Administration server.

Starting a Content Server in the Application Server

Both the Content client and the Content client (Classic) use the Portal Manager API. For making this API available (also for custom JSP pages), one of the Content servers must be started in the context of the application server.

Content server as a web application

If a web application has been generated for the Content server, the Content server runs in an application server. If the application server used supports the servlet standard 2.3, the Content server is started and stopped automatically together with the respective web application.

If the application server supports the servlet standard 2.2, the web application and the Content server are started and stopped separately. In this case, use the respective scripts for the Content server (see the following section). Make sure that the web application is started first.

Starting the Content server in the JSP engine

If the Content server runs in the context of a JSP engine that does not support web applications, the scripts located in the directory **{WCM installation directory}\tools** must be used for starting the server.

The following steps are required:

1. To set the class path required for the Content server, call the script **setPomaClasspath.bat** or **.sh**. This script is located in the directory **{WCM installation directory}\tools**.
2. Add the class path that was created by the script to the class path of the JSP engine.

3. Copy the supplied script **portalmanager.bat** or **.sh**, which is located in the directory **{WCM installation directory}\tools**, to the root directory of the WCM installation. Rename the script **{name of the Content server}.bat** or **.sh**.
4. In the script **{name of the Content server}.bat** or **.sh**, replace all placeholders `SERVERNAME` with the name of the Content server.

In order to start the Content server via the script, the JSP engine must already be running. In the configuration of the JSP engine used, the servlet mapping `'/servlet/*'` must be entered (see also the sample configuration of Resin in section “Modifying the File resin.conf” on page 175).

> **Note:** If you set up two or more Content servers on the same computer, you must use different instances for your JSP engine for executing the servers. After the installation of Livelink WCM Server, modify the default URL in the scripts for starting the Content servers according to the configuration of the JSP engine used.

For stopping the Content server, use the supplied script **shutdown_{name of the Content server}.bat** or **.sh**. Alternately, the server can be shut down via the Admin client.

Starting Search Servers

Content Miner Search servers

If the Content Miner Search server is located on the same host computer as the assigned Content server, the Search server is started and stopped automatically together with the Content server. In this case, it is also possible to start and stop the Search server via the system administration of the Admin client. For starting the Search server, the assigned Content server must at least be in run level 4 “Single user mode”. For detailed

information on how to proceed in the Admin client, refer to the Content Miner Manual

If the Content Miner Search server and the assigned Content server are located on different hosts, you can start the Search server by means of the script supplied. The start script **start_{name of the Search server}.bat** or **.sh** is located in the directory **{WCM installation directory}\contentminer\{name of the Search server}\server**.

Notes

- After a new installation, it may take some time to start the Search server as this requires initialization of the Readware server.
- When you start the Search server with the start script, the following message indicates that the server is up: `Press “Ctrl C” to halt`.
- When starting the Search server under Windows via the respective start script, it is no longer possible to stop this Search server via the Admin client.
- After assigning a Search server to a Content server and after deleting all Search servers of the WCM system, the respective Content server(s) must be restarted.

Under UNIX, you can also start and stop the Index and Query systems separately by means of scripts. For this purpose, the following scripts are provided:

- The script **start_{Search server name}_analyst.sh** starts the Index system only.
- The script **start_{Search server name}_query.sh** starts the Query system only.
- The script **stop_{Search server name}_analyst.sh** stops the Index system.

- The script **stop_{Search server name}_query.sh** stops the Query system.

Lucene Search servers

Lucene Search servers can only be installed on the same host computer as the assigned Content server. They are always started and stopped automatically together with the Content server.

Log Files and Error Files

For all installation and deinstallation processes, a log is written to the file **{WCM installation directory}\installation\installation.log**. Any errors that occur are written to the file **error.log**, which is located in the same directory.

All server actions are recorded in the log and error files of the respective servers.

If the parameter `-noredirect` is not set in the server start scripts (see table “Parameters in the server start scripts” on page 154), each server creates the files **{server name}_out.log** (for standard messages) and **{server name}_err.log** (for error messages) in the **log** directory.

In addition, the file **{server name}_com.log** is created in the **log** directory; communication errors are written to this file.

> **Note:** The log files of the servers can also be viewed via the Admin client (see Livelink WCM Server Administrator Manual).

The number and size of the log files can be configured via the parameters in the server start scripts (see table “Parameters in the server start scripts” on page 154).

CHAPTER 5
Configuring the Web Server

In Livelink WCM Server, objects are accessed via a URL in order to edit WCM objects with the Content client and to display generated pages by means of a browser. For this purpose, an HTTP server that maps the deployment system directories to the relevant URLs must be installed on the computer hosting the respective WCM server. Various components of the WCM system (Content client and Content client (Classic), dynamic deployment and InSite Editing, WebDAV, Secure Access) additionally require a JSP engine. The configurations of HTTP server and JSP engine must be adjusted to each other.

> **Note:** The process of the JSP engine must have write permissions for certain directories. For more information, refer to section “Write permissions for the directories of Livelink WCM Server” on page 117.

For providing HTTP services and for processing JSP scripts, different software products can be used. Web servers combine the functionalities of HTTP servers and JSP engines. They can also be realized by means of application servers or a combination of HTTP server plus JSP engine.

This chapter provides information on how to configure web servers for the different Livelink WCM Server components on the basis of selected examples. For this purpose, the following example configurations will be described:

- using the HTTP server Apache with the JSP engine Tomcat (see the following section)
- using the application server BEA WebLogic (see section “Scenario 2 – BEA WebLogic 8.1” on page 170)
- using the HTTP server Microsoft Internet Information Server with the JSP engine Resin (see section “Scenario 3 – MS Internet Information Server with Resin” on page 172)

To speed up loading the Content client, you can use the precompile script (see section “The Precompile Script” on page 188).

A separate section describes the configuration required for using Secure Access (see section “Configuring Secure Access” on page 189).

Notes

- This documentation can only describe the configuration of selected web servers for certain areas of application. For more information and configuration notes, refer to the documentation of the products that you use.
- For information on the versions of the tested HTTP servers and JSP engines that Livelink WCM Server works with, refer to the Release Notes.
- If the web container of the application server or the JSP engine used only supports the Servlet API 2.2, Unicode characters cannot be used in the metadata of the WCM objects. Thus, in this case, you must set the Content client and the website to Latin-1 encoding. The encoding for the Content client is specified during installation of the Content server running in the JSP engine, the encoding for the website is selected when the website is created.

- When installing the RDBMS Oracle under Windows 2000, the Apache HTTP server is installed automatically and registered as a service with “Automatic” startup. If you use a different HTTP/web server or a different Apache configuration, deactivate this service or deinstall the product.
- In order to avoid problems in connection with the Content server that runs in the context of the JSP engine or as a web application in an application server, the timeout value of the JSP engine should be set to the same value that Livelink WCM Server uses for automatically logging out users. This *Expiration interval* is set in the Administration server settings on the *Administration* tab.

  If the session timeout of the JSP engine is shorter than the expiration interval, more licenses may be in use than actually should be. The reason for this is that when the JSP session times out, the user must re-login to the Portal Manager API. The login to Livelink WCM Server, however, remains valid. In this case, two licenses are used until the first login times out in Livelink WCM Server.

  If the JSP session timeout is longer than the timeout of Livelink WCM Server, an `InvalidContextIdException` is thrown if a user has been inactive and Livelink WCM Server times out while the JSP session is still active. The JSP session timeout is set by means of the following parameter:

```
<web-app app-dir='d:/wcm/website' id='/'>
   <session-config>
      <session-timeout>120</session-timeout>
   </session-config>
</web-app>
```

Scenario 1 – Apache 2 with Tomcat

Prerequisites

- installed HTTP server Apache 2 (here: version 2.0.55 for Windows)
- installed Tomcat JSP engine (here: version 5.5.17)
- suitable JK plugin (jk-1.2.15, **mod_jk-apache-2.0.55.so** file)

  The JK plugin is responsible for the communication between Apache and Tomcat. For this purpose, a Tomcat-specific communication protocol is used.

  The JK plugin is available for download at: http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/.

> **Note:** The scenario described here assumes that Apache 2 and Tomcat are running on the same host computer. Apache 2 uses port 80 for the HTTP connections. Thus, this port may not be used by Tomcat.

Configuration steps

1. Stop Apache and Tomcat.
2. Use the Admin client to create a web application for the WCM servers. (For detailed information, refer to section “Generating a Web Application” on page 94.)

   As target directory of the generated WAR file, specify the directory **{Tomcat installation directory}\webapps**.

   In the following, it is assumed that the name of the web application is “wcm” and the name of the Content client is “cmsclient”.
3. Install the JK plugin. To do so, rename the file **mod_jk-apache-2.0.55.so** to **mod_jk.so** and copy the file to the directory **{Apache 2 installation directory}\modules**.

4. Apache 2 is configured in the **httpd.conf** file in the directory **{Apache installation directory}\conf**. In this file, add the following lines to the `LoadModule` section:

```
LoadModule     jk_module  modules/mod_jk.so
JkWorkersFile  conf/workers.properties
JkLogFile      logs/mod_jk.log
JkLogLevel     info
JkLogStampFormat  "[%a %b %d %H:%M:%S %Y]"
JkOptions      +ForwardKeySize +ForwardURICompat -ForwardDirectories
JkRequestLogFormat "%w %V %T"
JkMount  /wcm/* worker1
```

5. Create the file **workers.properties** in the directory **\{Apache 2 installation directory}\conf**. Enter the following as the content of the file:

```
worker.list=worker1
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
worker.worker1.lbfactor=50
worker.worker1.cachesize=10
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.recycle_timeout=300
```

6. Start the Apache server and afterwards the Tomcat server.

   During the start, the web application is automatically installed in the Tomcat server and started.

Using HTTPS connections to the Content client

You can use HTTPS for the connection to the Content client. As the configuration files of the Content client are loaded via an HTTPS request in this case, the webserver's certificate must be added to the truststore of the JSP engine.

Notes on Using Secure Access

- For general information on Secure Access, refer to section “Configuring Secure Access” on page 189.
- All URLs that correspond to the pattern specified for the Tomcat plugin in the **httpd.conf** file under `JkMount` (in the sample code, all URLs with the prefix “wcm/”) are forwarded by the Apache 2 server to the Tomcat server. The functions of the access control component Secure Access are executed completely in the Tomcat server. Thus, Secure Access is configured in the web application generated in step 2 (in the section `<filter>` of the **web.xml**) file.
- In order to use Secure Access, set the parameter `pathto` for the Secure Access filter in the file **web.xml** to the value `{Tomcat installation directory}\webapps\`.

Scenario 2 – BEA WebLogic 8.1

Prerequisite

Both the web server and the servlet engine of BEA WebLogic 8.1 are used.

Configuration steps

1. Stop WebLogic.
2. Use the Admin client to create a web application for the WCM servers. (For detailed information, refer to section “Generating a Web Application” on page 94.)

   If you do not generate the web application as the default application, the name of the web application (e.g. wcm) will become part of the URL used for accessing the WCM system.

   Example: http://wcmserver.company.example:7001/wcm

3. In the directory **{WebLogic installation directory}\user_projects\mydomain\applications**, create a subdirectory with the name of the web application.
4. Copy the WAR file generated in step 2 to the directory created in step 3 and extract the WAR file.

   After the WAR file has been extracted, the directory contains the subdirectories **\cmsclient**, **\htmlclient**, and **\WEB-INF**.
5. Delete the WAR file from the directory.
6. Use the Admin client to create websites. The following must be considered:

   - *Base URL:* In addition to the URL for the deployment system, specify the port number used by the WebLogic server. If you have not generated the web application as default application, insert the name of the web application in the URL.

     Example: http://wcmserver.company.example:7001/wcm
   - *Directory:* Specify the path to the directory created in step 3.

   For detailed information on creating websites, refer to the Livelink WCM ServerAdministrator Manual.
7. Only if the web application was generated as default application: Set the URI prefix of the web application to "/". To do so, create the file **weblogic.xml** in the directory **\{WebLogic installation directory}\user_projects\mydomain\applications\{name of the web application}\WEB-INF**. Enter the following contents:

```
<weblogic-web-app>
   <context-root>/</context-root>
</weblogic-web-app>
```

Using HTTPS connections to the Content client

You can use HTTPS for the connection to the Content client. As the configuration files of the Content client are loaded via an HTTPS request in this case, the webserver's certificate must be added to the truststore of the JSP engine.

Notes on using Secure Access

- For general information on Secure Access, refer to section “Configuring Secure Access” on page 189.
- In order to use Secure Access, set the parameter `pathto` for the Secure Access filter in the file **web.xml** to the value **{WebLogic installation directory}\user_projects\mydomain\applications**.

Scenario 3 – MS Internet Information Server with Resin

The configuration of the mappings of directories to URLs for the HTTP server MS Internet Information Server (IIS) and the according configuration of the JSP engine Resin will be described for the following **example**:

- The master Content server is installed on the computer with the host name “wcmserver.company.example”.
- The website “InternetSite” is created.
- On the master Content server, there are two deployment systems (Edit and QA) for this website.

  The data of the deployment systems is saved in the directory **\website** below the WCM installation directory. In this directory, two subdirectories for the Edit and QA deployment systems are created:

- Edit deployment system: **d:\wcm\website\InternetSite_edit**
- QA deployment system: **d:\wcm\website\InternetSite_qa**

A Content server is created as standard application with the name “wcm”. The file **wcm.war** is extracted to the directory **d:\wcm\website**.

In the Admin client, the following URLs are specified for accessing the pages generated by the deployment systems:

- Edit view: http://wcmserver.company.example/InternetSite_edit
- QA view: http://wcmserver.company.example/InternetSite_qa

Configuring MS Internet Information Server

To configure the URL mapping in IIS:

1. Start the Internet Services Manager.
2. In the tree on the left, select the computer hosting the WCM server server.
3. In the right window pane, mark the entry *Default web site* and choose *Properties* on the context menu.
4. On the *Home Directory* tab, specify the local path for the URL mapping in the field *Local Path*.

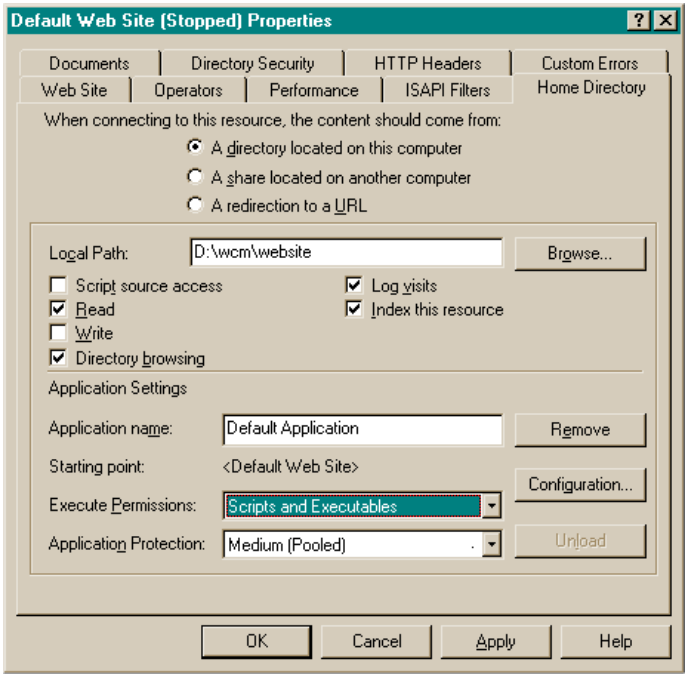


Fig. 20 – URL mapping in IIS

5. Confirm the settings by clicking the *OK* button.

Configuring Resin

In the following, the required steps for configuring the JSP engine are described, starting with registering the JSP engine with the HTTP server. The example refers to the JSP engine Resin and the HTTP server IIS (version 5.0) under Windows 2000.

Configuring Resin comprises the following steps:

1. Modifying the file **resin.conf** (see the following section)

2. Integrating Resin in IIS (see section “Integrating Resin in MS Internet Information Server” on page 177)
3. Starting Resin (see section “Starting Resin” on page 178)

Modifying the File resin.conf

Basically, Resin is configured via the file **resin.conf** in the directory **{Resin installation directory}\conf**.

The WCM server relevant to Resin is a web application that is configured via the **web.xml** file. This file is automatically created when the web application is generated and is located in the **\WEB-INF** directory of the web application (see section “Generating a Web Application” on page 94).

Resin is integrated in IIS via a plugin (file **isapi_srun.dll**). The plugin decides which requests (URL requests) are forwarded to Resin and which are processed by IIS in another way. The file **resin.conf** must contain the respective information.

Moreover, the base directory of the web application must be specified in the file **resin.conf**. The base directory can be determined either via the tag `<war-dir>` or the tag `<app-dir>`.

- `<war-dir>`: Specify the directory containing the WAR files that Resin is to extract and treat as a web application. This directory must be the base directory of the HTTP server.

  In this configuration, it is not possible to use the website directory itself as the directory for the so-called default application. (Resin expects the WAR file of a default application to be called **root.war** and extracts the web application to a directory **root**).

- `<app-dir>`: Directly specify a directory of the web application.

In this configuration, it is possible to use the website directory itself as the base directory for the web application. For this purpose, the file **wcm.war** must be extracted without Resin (e.g. by calling the command `jar –xf wcm.war` in the website directory). The following sample configuration is based on this approach.

Sample configuration

Note: The following sample configuration only describes the information required for the integration of Resin with IIS.

Insert a new `<web-app>` section in the file **resin.conf**. In a section, enter the base directory of the HTTP server as application directory, e.g. `<web-app app-dir='d:/wcm/website' id='/'>`.

In this section, the plugin is mapped. You can also determine the priority of the plugin for IIS here.

- Resin without Secure Access

```
<caucho.com>
<java compiler="internal" compiler-args=""/>
<!-- <iis priority='high'/> -->
<http-server class-update-interval='15'>
<srun port='6802'/>
<!-- VIP default -->
<web-app app-dir='d:/wcm/website' id='/'>
<!-- send to resin -->
<servlet-mapping url-pattern='*.jsp' servlet-name='plugin_match'/>
</web-app>
</http-server>
</caucho.com>
```

- Resin with Secure Access

```
<caucho.com>
<java compiler="internal" compiler-args=""/>
<!-- <iis priority='high'/> -->
<http-server class-update-interval='15'>
```

```
<srun port='6802'/>
<!-- VIP default -->
<web-app app-dir='d:/wcm/website' id='/'>
<!-- send to resin -->
<servlet-mapping url-pattern='/*' servlet-name='plugin_match'/>
<!-- don't send to resin -->
<servlet-mapping url-pattern='/secret/*' servlet-
name='plugin_ignore'/>
<!-- if asp pages shall not be checked by Secure Access, hide them
from resin! -->
<!-- <servlet-mapping url-pattern='*.asp' servlet-
name='plugin_ignore'/> -->
</web-app>
</http-server>
</caucho.com>
```

Alias for Content client:

If you have configured an alias for the Content client in the HTTP server, this alias must also be entered in the configuration of the JSP engine. For Resin, enter the following lines in the file **resin.conf**:

```
<path-mapping url-pattern="/cmsclient/*"
real-path="{WCM installation directory}/website/cmsclient" />
```

Integrating Resin in MS Internet Information Server

To integrate Resin in IIS:

1. Execute the file **{Resin installation directory}\bin\setup.exe**.
2. Make sure that IIS permits execution of the file **isapi_srun.dll** in the script directory.

   To do so, open the *Default Website Properties* dialog box, select the *Home Directory* tab, and click *Scripts and Executables* in the *Execute Permissions* drop-down list (see figure “URL mapping in IIS” on page 174).
3. Restart IIS.

Starting Resin

Finally, restart Resin.

WCM servers installed as Windows services

If the WCM servers are installed as services, we recommend that you also set up a service for the JSP engine. Execute the following command:

```
{Resin installation directory}\bin\httpd.exe –install-as resin
```

Afterwards, configure the service in such a way that the Administration server is started first and then Resin. Due to the entries made in the **web.xml** file, the Content server running in the context of the JSP engine or as a web application in an application server will also be started implicitly with Resin.

WCM servers not installed as Windows services

If the WCM servers are not installed as services, the JSP engine (and thus also the Content server running in the context of the JSP engine or as a web application in an application server) must be started by means of the following call after the start of the Administration server:

```
start "resin" {Resin installation directory}\bin\httpd.exe
```

Using HTTPS Connections to the Content Client

You can use HTTPS for the connection to the Content client. As the configuration files of the Content client are loaded via an HTTPS request in this case, the webserver's certificate must be added to the truststore of the JSP engine.

Notes on Using Secure Access

Note: For general information on Secure Access, refer to section “Configuring Secure Access” on page 189.

When using IIS as HTTP server, different authentication methods can be used when a user accesses a directory protected by Secure Access.

- base authentication: standard authentication via the login dialog box of the browser (see section “Configuring Base Authentication” on page 179)
- NTLM authentication (only under Windows when using IIS together with Internet Explorer): For logging in to the WCM system, the user account of the Windows operating system is used. Login to the WCM system is performed in the background without the user having to enter login information (see section “Configuring NTLM Authentication” on page 186).

Configuring Base Authentication

In the case of a base authentication, login to the WCM system is performed via a login dialog box in the browser. This presupposes that in the settings of the respective paths for Secure Access in the Admin client, the option *base* is selected under *Login method* (see Livelink WCM Server Administrator Manual).

In IIS, the base authentication can be configured as anonymous access or as access with user name and password. This does not make any difference for the user when logging in to the WCM system. In both cases, the user data for the WCM system must be entered in the login dialog box of the browser.

Anonymous access

To configure anonymous access for base authentication:

1. Start the Internet Services Manager.
2. In the tree on the left, select the computer hosting the WCM server server.
3. In the right window pane, mark the entry *Default web site* and choose *Properties* on the context menu.
4. Go to the *Directory Security* tab. In the *Anonymous access and authentication control* area, click the *Edit* button.
5. In the *Authentication Methods* dialog box, select the *Anonymous access* check box.

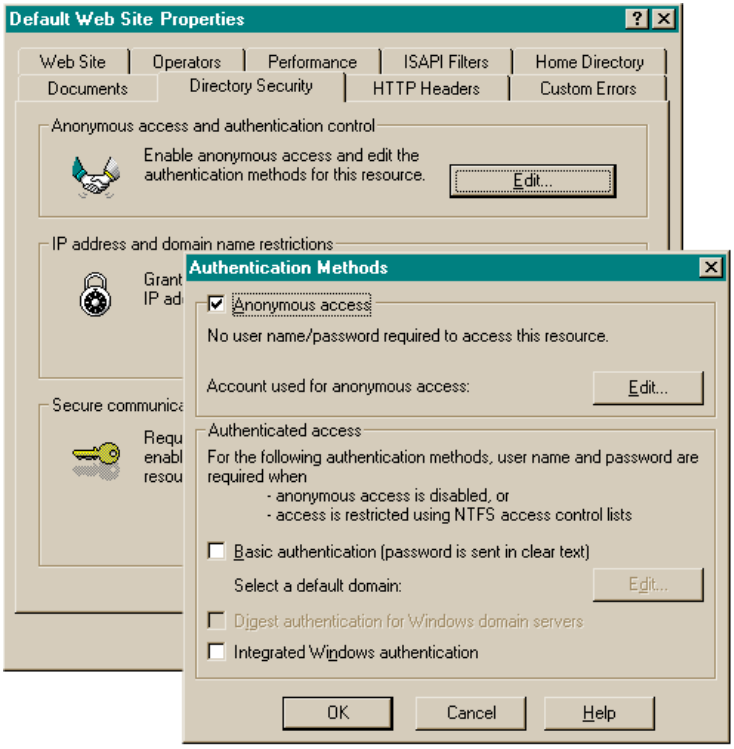


Fig. 21 – IIS configuration for base authentication with anonymous access

Access via user name and password

This configuration of the base authentication presupposes that the file **VIPSecure.dll** is configured as ISAPI filter for the protected directories. This file is located on the WCM CD in the directory **\VipSecure\Binaries**.

Configure the file **VIPSecure.dll** as ISAPI filter for the protected directories. Proceed as follows:

1. Start the Internet Services Manager.
2. In the tree on the left, mark the computer hosting the WCM server and choose *Properties* on the context menu.
3. In the *Properties* dialog box, click the *Edit* button in the *Master Properties* area of the *Internet Information Services* tab.
4. In the *WWW Service Master Properties* dialog box, go to the *ISAP Filters* tab. Click the *Add* button.
5. In the *Filter Properties* dialog box, enter the name of the filter and the path to the file **VIPSecure.dll**.

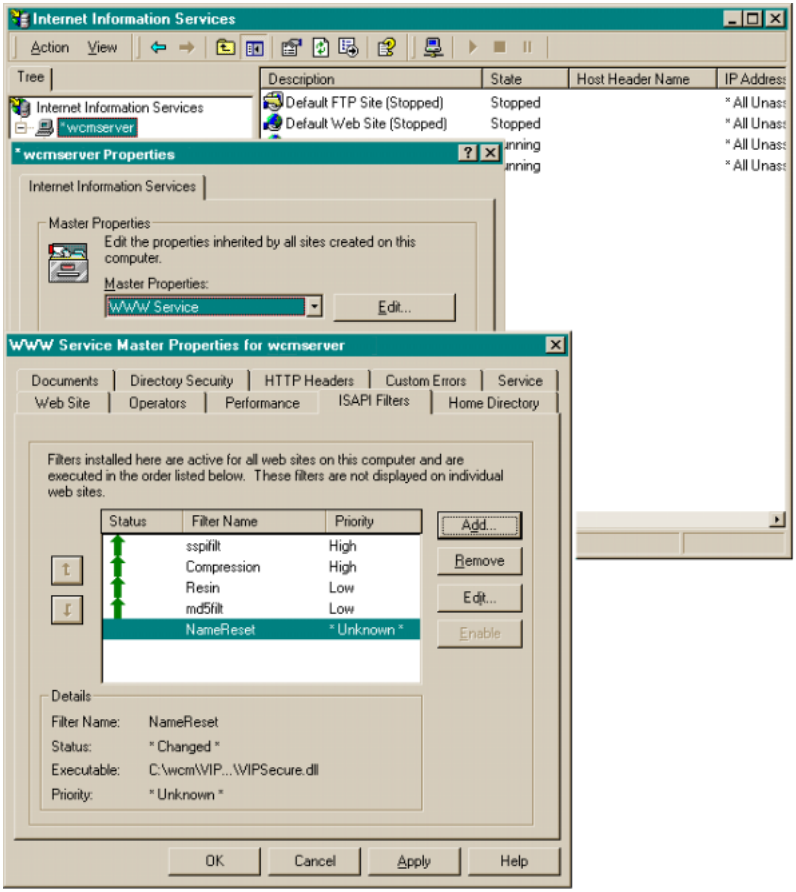


Fig. 22 – Configuring the VIPSecure.dll as a ISAPI filter (IIS)

The DLL maps the name and password which the user enters in the browser for accessing protected files to the login information of a Windows user. This login information is written to the file **vipsecure.ini**.

1. Create the file **vipsecure.ini**.
2. Enter the following text in the file:

```
[ntlm]
user=
pwd=
```

3. Copy the file to a Windows directory, e.g. **C:\WINNT**. This directory must be accessible via the Windows system environment variable PATH.

To configure access via user name and password for base authentication:

1. Start the Internet Services Manager.
2. In the tree on the left, select the computer hosting the WCM server server.
3. In the right window pane, mark the entry *Default web site* and choose *Properties* on the context menu.
4. Go to the *Directory Security* tab. In the *Anonymous access and authentication control* area, click the *Edit* button.
5. Only select the check box *Basic authentication (password is sent in clear text)*.

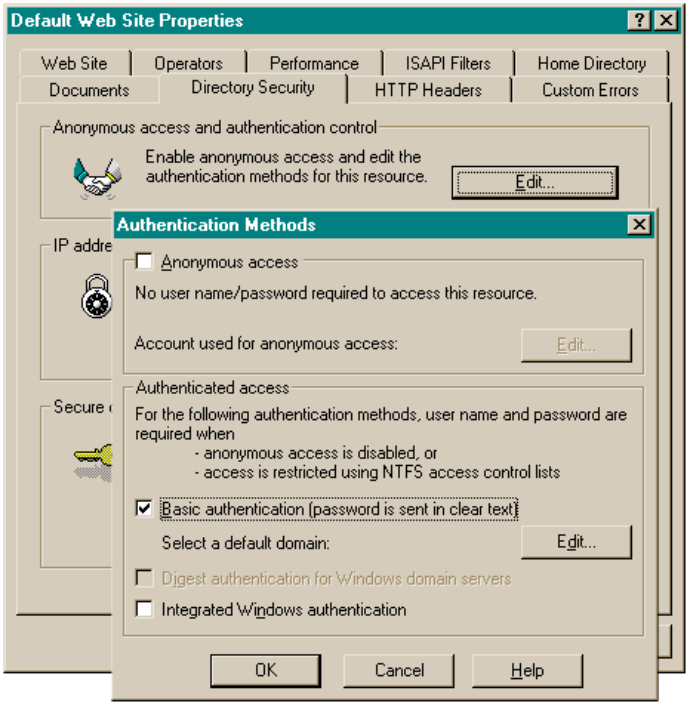


Fig. 23 – IIS configuration for base authentication with name and password

6. In the file **vipsecure.ini**, enter the login data of a user who is registered on this computer and who has according access rights in the file system.

   The file **VipSecure.dll** maps the login data the user enters when accessing a file protected by Secure Access to this login data.

Configuring NTLM Authentication

The procedure NTLM can be used as an alternative to the base authentication. If a user requests a protected page, the login information is read from the operating system and login to the WCM system is performed in the background.

Prerequisites

- In the settings of the respective paths for Secure Access in the Admin client, the option *ntlm* must be selected under *Login method* (see Livelink WCM Server Administrator Manual).
- If you use an RDBMS for saving the user data, the user IDs for the WCM system configured in the Admin client must exist as Windows user names and contain the name of the Windows domain, e.g. “COMPANYDOMAIN\jstein”. If an LDAP directory service is used, the domain can be entered in a separate field in the user settings.
- In the settings of the WCM users in the Admin client, the *Trusted login* check box must be selected.
- The **VipSecure.dll** may **not** be used.

Configuring IIS and Internet Explorer

To configure NTLM:

1. Start the Internet Services Manager.
2. In the tree on the left, select the computer hosting the WCM server server.
3. In the right window pane, mark the entry *Default web site* and choose *Properties* on the context menu.
4. Go to the *Directory Security* tab. In the *Anonymous access and authentication control* area, click the *Edit* button.

5. Select the check boxes *Anonymous Access* and *Integrated Windows authentication*.

> **Note:** If you only select the *Integrated Windows authentication* check box, internal URL connections of Livelink WCM Server and Secure Access will fail. In this case, checking WCM objects out and in with the Download applet, statification, and processing of the redirect and welcome page of Secure Access will not work.

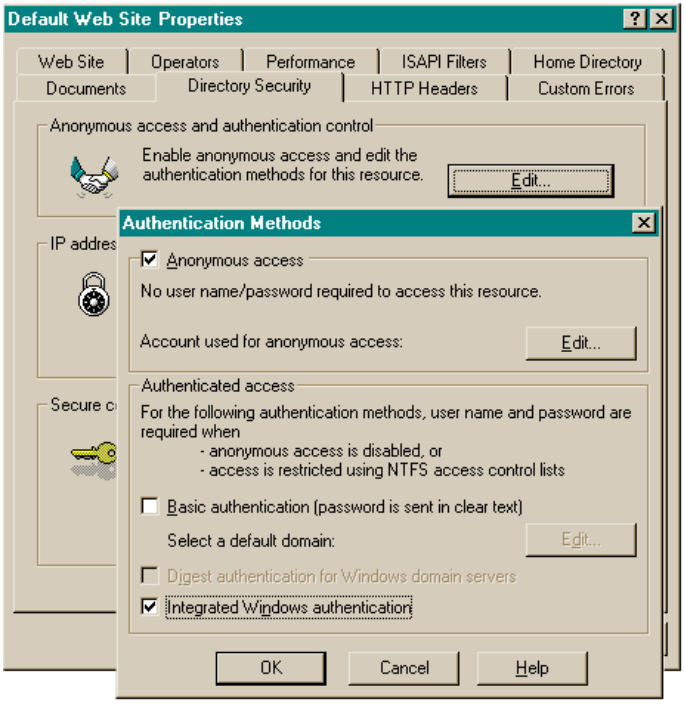


Fig. 24 – IIS configuration for NTLM authentication

In the settings of the users' browser, automatic login must be activated. In Internet Explorer, this is done via *Tools* → *Internet Options* → *Security* tab → *Custom Level* button → *User Authentication* → radio button *Automatic logon with current username and password*.

The Precompile Script

After the installation, the directory **{WCM installation directory}\tools** contains the scripts **precompile.bat** and **precompile.sh** which enable you to precompile the Content client. This makes the first access to the Content client faster.

For executing the script, the Admin server, the master Content server, the JSP engine, and the Content server running in the context of the JSP engine or as a web application in an application server must be started.

This is how you call the script:

```
precompile {application directory of the JSP engine} {JSP base URL}
```

The individual components consist of the following:

- `{application directory of the JSP engine}`: Enter the application directory you have defined for the Content client in the configuration of the JSP engine.

  Example: `d:/wcm/website/cmsclient`

- `{JSP base URL}`: URL for accessing the application directory specified. The mapping of directories to URLs is set in the HTTP server configuration.

  Example: `http://wcmserver.company.example/cmsclient`

The script should be run after *every* restart of the JSP engine, before the users start working with the Content client.

Configuring Secure Access

The deployment systems of a Content server generate HTML files, JSP pages, etc. in the configured directories. These directories can be accessed via an HTTP server and a web browser.

In order to protect the generated files from unauthorized access, e.g. via the Internet, you can arrange for certain directories or all directories in a deployment system to be monitored by Secure Access. This offers the possibility to transfer the access control used for editing a WCM object in the Content client to the web server or JSP engine. As a result, object-specific access control is possible for all objects in a website during productive operation (i.e. in the Internet as well).

It is not necessary to protect dynamic deployment systems with Secure Access. The servlets used for the dynamic deployment are responsible for access control.

How Secure Access works

Secure Access is implemented as a servlet or filter and runs in the JSP engine. When a page is requested from the HTTP server via a browser and the HTTP server forwards the request to the JSP engine, Secure Access intercepts this request and checks whether it is addressed to one of the configured protected directories. If this is the case, the user must enter user ID and password for access to the WCM system in a login dialog box in the browser.

As an alternative to logging in via the browser, the login information of the Windows operating system may also be used for authenticating the user. This is possible if you operate the WCM system in a Windows domain and use MS Internet Information Server as the web server and MS Internet Explorer as browser. To enable this login procedure, you must select the login method *ntlm* in the settings of the Secure Access paths. For information on configuring NTLM, see section “Configuring NTLM Authentication” on page 186.

The login data is checked by the Admin server of the WCM system. If the login of the user has been successful, the pages can be displayed via the HTTP server and the user can view the object in the browser.

Required configuration

The following settings must be made in order to use Secure Access:

- activating Secure Access and adding the paths for Secure Access in the Admin client (see the following section)
- registering the JSP engine with the HTTP server (see sections “Scenario 1 – Apache 2 with Tomcat” on page 168 and “Scenario 3 – MS Internet Information Server with Resin” on page 172)
- configuring the Secure Access parameters (see section “Common Parameters for the Secure Access Servlet and the Secure Access Filter” starting on page 197) and activating the entries in the **web.xml** file

It is possible to modify the HTTP requests for protected pages by means of a Java class. This way, you can, for example, allow automatic login for users on the basis of their IP address. See section “Post-Processing Requests for Protected Pages” on page 209.

Settings in the Admin Client

The access control component Secure Access must be activated for the servers whose deployment systems are used to publish the protected files. This presupposes that these deployment systems have already been created. To activate Secure Access for a server, the following settings must be made in the Admin client:

Activating Secure Access in the server settings

To activate Secure Access in the server settings:

1. Open the server settings via *Configuration* → *Servers* → *{server name}*.
2. On the *Secure Access tab*, select the *Secure Access activated* check box.
3. Enter the name of the HTTP server used in the field *Web server* on the *Secure Access* tab. This name must correspond to the name entered in the parameter `profile` when configuring the JSP engine (see section “Common Parameters for the Secure Access Servlet and the Secure Access Filter” on page 197).

Adding paths for Secure Access

To protect the files in a deployment system directory, the path to this directory must be registered.

To add a path for Secure Access:

1. Select *Configuration* → *Servers* → {*server name}* → *Secure Access assigned paths*.
2. Specify the directories to be protected by Secure Access.

Notes:

For detailed information on configuring Secure Access with the Admin client, refer to the Livelink WCM Server Administrator Manual (chapter “Managing Servers”).

After adding paths for Secure Access via the Admin client, the JSP engine must be restarted for the new configuration to take effect.

Configuring the HTTP Server for Secure Access

The HTTP server must be configured in such a way that all incoming requests are forwarded to the JSP engine.

Scenario 1 – Apache with Tomcat

All URLs that correspond to the pattern specified for the Tomcat plugin in the file **httpd.conf** under `JkMount` are forwarded from the Apache server to the Tomcat server. The functions of the access control component Secure Access are executed completely in the Tomcat server.

Scenario 2 – BEA WebLogic 8.1

In order to use Secure Access with BEA WebLogic 8.1, no special configuration settings are necessary.

Scenario 3 – MS Internet Information Server with Resin

For information on registering Resin with IIS, refer to section “Integrating Resin in MS Internet Information Server” on page 177. There, the general procedure of integrating the JSP engine in the HTTP server is described, i.e. this is not a Secure Access-specific configuration.

Note: For information on configuring the different login methods (“base” and “ntlm”), refer to section “Notes on Using Secure Access” starting on page 179.

Configuring the JSP Engine for Secure Access

The parameters for the Secure Access servlet or the Secure Access filter are configured in the settings of the JSP engine used.

> **Note:** The Secure Access servlet and the Secure Access filter offer the same features. To be able to use the filter – for which configuration is easier – the JSP engine must support the JSP specification 2.3. For the servlet, 2.2 is sufficient.

Secure Access is automatically configured in the file **web.xml** when the Content server is set up as a web application. In most cases, the parameter `pathto` must be modified. The value of this parameter must be the base directory of the HTTP server.

For the Secure Access servlet, the following entries are generated in the **web.xml** file. (Please note that the parameters, such as server name, directories, and port numbers, depend on the information specified during generation of the web application.)

```
<!--
<servlet>
      <servlet-name>AccessServlet</servlet-name>
      <servlet-class>de.gauss.vip.vipsecure.AccessServlet
         </servlet-class>
      <init-param>
         <param-name>character-encoding</param-name>
         <param-value>UTF-8</param-value>
      </init-param>
      <init-param>
         <param-name>pathfrom</param-name>
         <param-value></param-value>
      </init-param>
      <init-param>
         <param-name>pathto</param-name>
         <param-value>d:/wcm/website</param-value>
      </init-param>
      <init-param>
         <param-name>usepath</param-name>
         <param-value>false</param-value>
      </init-param>
```

```
      <init-param>
         <param-name>viphost</param-name>
         <param-value>wcmserver.company.example</param-value>
      </init-param>
      <init-param>
         <param-name>vippport</param-name>
         <param-value>5008</param-value>
      </init-param>
      <init-param>
         <param-name>httpport</param-name>
         <param-value>5009</param-value>
      </init-param>
      <init-param>
         <param-name>secure</param-name>
         <param-value>false</param-value>
      </init-param>
      <init-param>
         <param-name>profile</param-name>
         <param-value>wcm</param-value>
      </init-param>
      <init-param>
         <param-name>loglevel</param-name>
         <param-value>0</param-value> <!-- 191 -->
      </init-param>
      <init-param>
         <param-name>realm</param-name>
         <param-value>WCM</param-value>
      </init-param>
      <init-param>
         <param-name>logdir</param-name>
         <param-value>d:/wcm/log</param-value>
      </init-param>
      <init-param>
         <param-name>logname</param-name>
         <param-value>servlet_1</param-value>
      </init-param>
</servlet>
-->

<!--
<servlet-mapping>
      <servlet-name>AccessServlet</servlet-name>
      <url-pattern>/*</url-pattern>
</servlet-mapping>
-->
```

For the Secure Access filter, the following entries are generated in the file **web.xml**, e.g. for the JSP engine Resin:

```
<!--
<filter>
      <filter-name>AccessFilter</filter-name>
      <filter-class>de.gauss.vip.vipsecure.AccessFilter</filter-class>
         <init-param>
            <param-name>character-encoding</param-name>
            <param-value>UTF-8</param-value>
         </init-param>
         <init-param>
            <param-name>pathfrom</param-name>
            <param-value></param-value>
         </init-param>
         <init-param>
            <param-name>pathto</param-name>
            <param-value>d:/wcm/website</param-value>
         </init-param>
         <init-param>
            <param-name>usepath</param-name>
            <param-value>false</param-value>
         </init-param>

         <init-param>
            <param-name>viphost</param-name>
            <param-value>wcmserver.company.example</param-value>
         </init-param>
         <init-param>
            <param-name>vippport</param-name>
            <param-value>5008</param-value>
         </init-param>
         <init-param>
            <param-name>httpport</param-name>
            <param-value>5009</param-value>
         </init-param>
         <init-param>
            <param-name>secure</param-name>
            <param-value>false</param-value>
         </init-param>
         <init-param>
            <param-name>profile</param-name>
            <param-value>wcm</param-value>
         </init-param>
         <init-param>
            <param-name>loglevel</param-name>
            <param-value>0</param-value> <!-- 191 -->
```

```
         </init-param>
         <init-param>
            <param-name>realm</param-name>
            <param-value>WCM</param-value>
         </init-param>
         <init-param>
            <param-name>logdir</param-name>
            <param-value>d:/wcm/log</param-value>
         </init-param>
         <init-param>
            <param-name>logname</param-name>
            <param-value>filter_1</param-value>
         </init-param>
</filter>
-->

<!-- for IIS usage
      <init-param>
         <param-name>pattern1</param-name>
         <param-value>asp</param-value>
      </init-param>
      <init-param>
         <param-name>servlet1</param-name>
         <param-value>redirect</param-value>
      </init-param>
      <init-param>
         <param-name>redirect-url</param-name>
         <param-value>http://wcmserver.company.example/secret
            </param-value>
      </init-param>
-->

<!--
<filter-mapping>
      <filter-name>AccessFilter</filter-name>
      <url-pattern>/*</url-pattern>
</filter-mapping>
-->
```

Common Parameters for the Secure Access Servlet and the Secure Access Filter

In the following, the entries that can be configured for both the servlet and the filter will be explained.

Parameter `servlet-name` or `filter-name`

Use this parameter to specify whether the servlet or the filter is to be used. Possible entries:

- servlet for Secure Access

```
<servlet-name>AccessServlet</servlet-name>
<servlet-class>de.gauss.vip.vipsecure.AccessServlet
   </servlet-class>
```

- filter for Secure Access

```
<filter-name>AccessFilter</filter-name>
<filter-class>de.gauss.vip.vipsecure.AccessFilter
   </filter-class>
```

Parameter `character-encoding`

Example:

```
<init-param>
      <param-name>character-encoding</param-name>
      <param-value>UTF-8</param-value>
</init-param>
```

You can use this parameter to set the encoding for transferring the Secure Access parameters.

If UTF-8 was selected for the Content client during installation of the Content server running in the JSP engine, the servlet/filter of Secure Access must also be set to `UTF-8`. Alternately, you can specify `ISO-8859-1` for Latin-1 encoding. If this parameter is not set, Latin-1 encoding is used.

> **Note:** The use of UTF-8 (Unicode) must be configured consistently for all components of Livelink WCM Server. Please also refer to the notes on using Unicode in chapter “Concepts” of the Livelink WCM Server Administrator Manual.

Parameters `pathfrom` and `pathto`

Example:

```
<init-param>
      <param-name>pathfrom</param-name>
      <param-value></param-value>
</init-param>
<init-param>
      <param-name>pathto</param-name>
      <param-value>d:/wcm/website</param-value>
</init-param>
```

This is an additional replacement mechanism for using path information instead of URLs.

- In case the parameter `usepath` has the value `false`: In the parameter `pathto`, enter the base directory according to the configuration of the HTTP server. (For Apache, this corresponds to the parameter `DocumentRoot` or the alias – if configured; for Tomcat, this corresponds to the path to the directory **\webapps**.) The parameter `pathfrom` should remain empty.
- In case the parameter `usepath` has the value `true`: If errors occur in the path information – e.g. caused by virtual directories – you can

correct the first part of the path information by setting both parameters.

Parameter `usepath`

Example:

```
<init-param>
      <param-name>usepath</param-name>
      <param-value>false</param-value>
</init-param>
```

Indicates whether requests are processed on the basis of URLs or mapped path information. Possible values:

- `true`
  Path information is used instead of URLs.
- `false`
  URLs are used.

Which option you use, depends on whether the JSP engine supports the use of path information. If this applies to all object types, you can enter `true` here.

Server parameters (`viphost`, `vippport`, `httpport`, `secure`)

Example:

```
<init-param>
      <param-name>viphost</param-name>
      <param-value>wcmserver.company.example</param-value>
</init-param>
<init-param>
      <param-name>vippport</param-name>
      <param-value>5008</param-value>
</init-param>
<init-param>
      <param-name>httpport</param-name>
      <param-value>5009</param-value>
</init-param>
```

```
<init-param>
      <param-name>secure</param-name>
      <param-value>false</param-value>
</init-param>
```

By means of these parameters, you specify the connection to a server. Enter the name of the host computer and the ports for reaching the server. The parameter `secure` specifies whether the connection is to be established via SSL.

> **Note:** These entries must correspond to the configuration in the Admin client. The parameters for the connection to the server are specified in the settings of the server's WCM pool.

Parameter `profile`

Example:

```
<init-param>
      <param-name>profile</param-name>
      <param-value>wcm</param-value>
</init-param>
```

This parameter contains the name of the web server as specified in the Admin client on the *Secure Access* tab of the server settings (see section "Activating Secure Access in the server settings" on page 191).

Parameters for the Secure Access log (`loglevel`, `logdir`, `logname`)

By default, the messages of Secure Access are logged in the file **vipsecure_servlet.log** or **vipsecure_filter.log** in the directory **{WCM installation directory}\log**. This log contains all requests by the web server to Livelink WCM Server and the according responses. By means of the log parameters, you can control the logging.

Notes:

The user under whose ID the web server is running must possess read and write permissions for the directory to which the log files are written (see section“Write permissions for the directories of Livelink WCM Server” on page 117). This applies to both Windows and UNIX.

In addition to this log, the requests and responses between Secure Access and the respective server can also be logged in the log file of the server. Use the parameter `-Dvip.vipsecure.debug=true` in the server start scripts to switch on logging (see table “Parameters in the server start scripts” on page 154).

- Parameter `loglevel`

  Example:

```
<init-param>
<param-name>loglevel</param-name>
   <param-value>24</param-value>
</init-param>
```

  The log level controls the detailedness of the Secure Access log. This entry is optional. The following log levels can be specified:

Table 21 – Log levels of Secure Access

Log level	Explanation
8	all error messages
16	all warnings

Log level	Explanation
32	all information messages, e.g. about requests and associated responses
64	detailed information
0	nothing

To combine different log levels, simply add up the numbers. If, for example, all messages with the levels “warning” (16) and “error” (8) are to be written to the log, set the `loglevel` value to 24. The default value is 0. A great value, such as 191 or 255, should only be used in exceptional cases, e.g. for locating errors.

- Parameter `logdir`

  Example:

```
<init-param>
   <param-name>logdir</param-name>
   <param-value>d:/wcm/log</param-value>
</init-param>
```

  Directory for saving the log with the Secure Access messages

- Parameter `logname`

  Example:

```
<init-param>
   <param-name>logname</param-name>
   <param-value>servlet-1</param-value>
</init-param>
```

  You can use this parameter to define an appendix for the default name of the Secure Access log. In this example, the name of the log would be **vipsecure_servlet-1.log**.

This parameter is optional. Specifying a name appendix might, for example, be recommendable if several Secure Access servlets or Secure Access filters are running in a JSP engine (or web server). This way, the messages of the different servlets and filters can be logged in separate files.

Parameter `realm`

Example:

```
<init-param>
      <param-name>realm</param-name>
      <param-value>WCM</param-value>
</init-param>
```

The text specified here is displayed in the login dialog box of the browser when a user logs in to access a protected directory.

Parameters `redirect-dir` and `redirect-url`

Example:

```
<init-param>
      <param-name>redirect-dir</param-name>
      <param-value>d:/wcm/website/secret</param-value>
</init-param>
<init-param>
      <param-name>redirect-url</param-name>
      <param-value>http://wcmserver.company.example/secret</param-value>
</init-param>

<init-param>
      <param-name>pattern1</param-name>
      <param-value>asp</param-value>
</init-param>
<init-param>
      <param-name>servlet1</param-name>
      <param-value>redirect</param-value>
</init-param>
```

Use these parameters to specify a directory and a URL for a special redirection, e.g. for ASP files. The parameters `redirect-dir` and `redirect-url` are only used if the servlet “redirect“ is specified for a pattern (e.g. `asp`).

Important!

The `redirect-url` specified must not be monitored by Secure Access! For Resin and IIS, the mapping for the plugin must be switched off explicitly, e.g. in the file **resin.conf** by an according URL mapping: `<servlet-mapping url-pattern='/secret/*' servlet-name='plugin_ignore'/>`

Moreover, the URL must not be directly accessible over the Internet. Configure your web server accordingly.

- `redirect-dir` (optional) – The files (whose type was specified by means of the parameter `patternX`) are copied with random file names to this directory in order to be processed, e.g. by the DLL for ASP pages.
- `redirect-url` – URL for accessing this directory. If the parameter `redirect-dir` was set, the HTTP server must be configured in such a way that the specified URL maps this directory. If no directory was specified in the parameter `redirect-dir`, the HTTP server must be configured in such a way that this URL maps the original directories of the files. In this case, the files are processed in the original directories.
- `pattern` and `servlet` – see section “Parameters `pattern` and `servlet`” on page 206

Parameter `url-pattern`

Example:

```
<servlet-mapping>
      <servlet-name>AccessServlet</servlet-name>
      <url-pattern>/*</url-pattern>
</servlet-mapping>

<filter-mapping>
      <filter-name>AccessFilter</filter-name>
      <url-pattern>/*</url-pattern>
</filter-mapping>
```

Use this parameter to specify the URLs that are to be protected by Secure Access. The following values are possible:

- `/*`
  All URLs accessible via this HTTP server.
- `/{name of the deployment system directory}/*`
  All URLs referencing the directories configured as paths for Secure Access in the Admin client. If you have defined several paths for Secure Access in the Admin client, make a separate entry for each directory.
- `'*.xyz'`
  All URLs referencing files with the file extension specified

Notes:

The different values cannot be combined with each other.

The URL mapping only works for requests that are forwarded to the JSP engine. In particular, if the HTTP server communicates with the JSP engine by means of a plugin, filter mappings are usually ignored by the plugin. Therefore, the JSP engine (and thus Secure Access) does not see these requests. In this case, you must additionally configure the plugin in such a way that the respective requests are forwarded to the JSP engine.

Additional Parameters for the Secure Access Servlet

If you use the Secure Access servlet, you must also provide mappings for all other servlets already configured in the JSP engine. If the Secure Access servlet processes all files in a protected directory (parameter `<servlet-mapping url-pattern='/{name of the deployment system}/*'`), some file types must be forwarded to other servlets afterwards.

> **Warning!**
>
> Manufacturer-specific servlets, such as the `com.caucho.server.http.FileServlet`, might have separate caching mechanisms that circumvent the security mechanisms of the Secure Access servlet.
>
> These servlets may not be configured as `servletX` in the pattern/servlet pairs described below. If their use is inevitable, the caching mechanism must be disabled (e.g. for Resin by means of `<cache enable='false'/>`). Otherwise, protected pages might be available to unauthorized users via the cache.

In the following, the entries that must be made for the servlet mapping will be explained.

Parameters `pattern` and `servlet`

Example:

```
<init-param>
      <param-name>pattern1</param-name>
      <param-value>jsp</param-value>
</init-param>
<init-param>
      <param-name>servlet1</param-name>
      <param-value>resin-jsp</param-value>
</init-param>
```

Use the parameter `patternX` to specify the extension of the files (without dot) which are to be forwarded to another servlet after having been processed by Secure Access.

Possible values for `servletX`:

- `'{name of the servlet}'`
  the name of the servlet. The value of this parameter depends on the JSP engine used.

  For Resin, the name of the JSP servlet is configured in the **app-default.xml** file which is located in the directory **{Resin-installation directory}\conf**. Versions earlier than and including 3. 0.8 use the default name “jsp”, later versions use the default name “resin-jsp”.

- `'redirect'`
  A separate directory and a special URL are to be used for forwarding. This may, for example, be required for processing ASP pages, which IIS performs by means of a special DLL. Directory and URL are specified in the parameters `redirect-dir` and `redirect-url`.

Configure further pattern/servlet pairs according to your needs.

Parameter `directory-servlet`

Example:

```
<init-param>
      <param-name>directory-servlet</param-name>
      <param-value>com.caucho.server.http.DirectoryServlet</param-value>
</init-param>
```

If you configure this parameter, the associated subdirectories are displayed as usual when a directory is opened in the browser.

Parameter `setcontentlength`

Example:

```
<init-param>
      <param-name>setcontentlength</param-name>
      <param-value>true</param-value>
</init-param>
```

By means of this parameter, you determine whether the content length (the size of the object returned by the server) is to be set by Secure Access. Setting this parameter is optional, the default value is `true`.

> **Note:** If you use the application server IBM WebSphere, set this parameter to `false`. The content length is set by IBM WebSphere.

Parameter `welcome-file-list`

Example:

```
<init-param>
      <param-name>welcome-file-list</param-name>
      <param-value>index.htm,index.html,index.jsp</param-value>
</init-param>
```

By means of this parameter, you can specify a list of files that Secure Access is to consider when processing a request for a directory. If the URL to be processed belongs to a directory, Secure Access checks whether one of the files specified in this parameter is contained in the directory and returns the content of the first file found this way.

Post-Processing Requests for Protected Pages

The directory **{WCM installation directory}\examples\vipsecure** contains the Java classes `IPTranslator` and `DomainTranslator`. By means of these classes, the HTTP requests for protected pages can be modified before they are sent to the WCM system. This way, the user name in the request can be set according to the IP address, for example. On the basis of this information, automatic authentication is possible. Another application example is the removal of the domain names contained in user IDs. By editing the class `IPTranslator` or `DomainTranslator`, you can determine in which way the requests are modified.

After the classes have been integrated in the JSP engine, they are called whenever a protected page is requested. From the request, information, such as the absolute path of the requested page, user, password, IP address, and context ID, are read. This information can be modified afterwards.

To configure post-processing of HTTP requests:

1. Adapt the supplied Java class `IPTranslator` or `DomainTranslator` to your requirements.

   Information on how to do this is contained in the comments of the classes.

2. Compile the changed Java class.
3. Make the classes available for the JSP engine by entering it in the class path of the JSP engine.

4. Add the following parameters to the configuration of the Secure Access servlet/filter.

```
<init-param>
       <param-name>translator</param-name>
       <param-value>{package name}.{class name}</param-value>
</init-param>
```

5. Restart the JSP engine.

CHAPTER 6

Upgrading Livelink WCM Server

The Upgrade option in the installation program enables you to transfer an existing WCM installation to a higher version and to adapt the existing website data to the new data structure.

In this chapter, the following topics are introduced:

- upgrade via the graphical user interface (see the following section)
- upgrade via the console (see section “Upgrade via Console” on page 221)
- required steps after an upgrade (see section “Steps Required after the Upgrade” on page 225)

Upgrade via the Graphical User Interface

Upgrading Livelink WCM Server is executed in two steps:

1. Adapting the configuration and copying the required files (see the following section)
2. Adapting the table structure and contents in the database used (see section “Upgrading the Data Storage” on page 217)

If you want to exchange the database which is used for storing website data, export the websites from the WCM system first. Use the Export/Import tool for this purpose. After this, upgrade the version. Add a pool for the new database and import the website to the upgraded WCM system, specifying the new pool for the import.

Notes:

You can only delete a website if the website version corresponds to the version of the WCM system.

For information on using the Export/Import Tool, refer to the Livelink WCM Server Administrator Manual (chapter “Managing Websites”).

Upgrading the Version

Notes

- The individual components of a distributed WCM system must be upgraded in the following order: first the master Admin server must be upgraded, then the proxy Admin servers (if existing). After this, the master Content server and any proxy servers are upgraded.
- In a distributed WCM system, a separate upgrade must be performed for each installation directory containing components of the WCM system. All components located in the directory will be upgraded at the same time.
- Before performing an upgrade, you must back up the data of the WCM system. For detailed information on the backup, refer to the Livelink WCM Server Administrator Manual.

  If you use the Oracle RDBMS, you should also update the statistics.

  If you use an LDAP-based user administration, perform a data backup for the LDAP directory service.

- Content servers can only access website data with a data structure that corresponds to their WCM version. A server of version 9.2.1, for example, cannot boot websites whose data storage has already been upgraded to version 9.5.

Prerequisites

- The WCM system to be upgraded has version 8.1.1 or higher.
- The servers to be upgraded are in the run level “Server down“.
- Make sure that no processes are accessing the files of the installed WCM system.
- Upgrading bases on copying directories. Thus, additional storage space is required in the file system for performing the upgrade.

Procedure

To upgrade the version by means of the graphical user interface of the installation program:

1. Start the installation program (see section “Starting the Installation” on page 69), and select the installation directory of the existing WCM system.
2. In the *Type of installation* dialog box, click the *Version upgrade* radio button and click the *Next* button.

   In the *Upgrade information* dialog box, the current version of the WCM system and the future version (after a successful upgrade) are displayed.

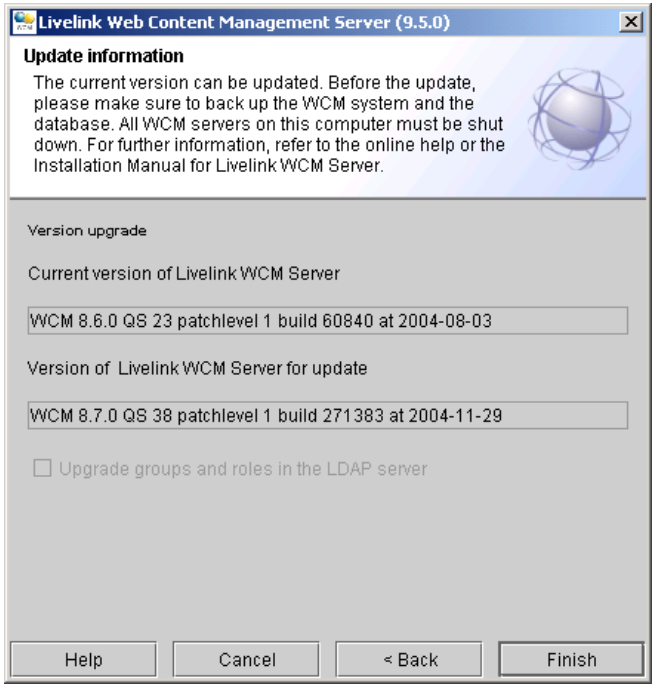

Livelink Web Content Management Server (9.5.0)
Update information
The current version can be updated. Before the update, please make sure to back up the WCM system and the database. All WCM servers on this computer must be shut down. For further information, refer to the online help or the Installation Manual for Livelink WCM Server.
Version upgrade
Current version of Livelink WCM Server
WCM 8.6.0 QS 23 patchlevel 1 build 60840 at 2004-08-03
Version of Livelink WCM Server for update
WCM 8.7.0 QS 38 patchlevel 1 build 271383 at 2004-11-29
Upgrade groups and roles in the LDAP server
Help
Cancel
< Back
Finish

Fig. 25 – Upgrading the version

3. If you used version 8.1 with an LDAP-based user administration in the past, the LDAP data structure may have to be modified. For this purpose, select the check box *Upgrade groups and roles in the LDAP server*.
4. Click the *Finish* button.

   All servers located in the specified directory will be upgraded at the same time.

   First, the upgrade program makes a backup copy of the directory **{WCM installation directory}\config**. Afterwards, the configuration files are adapted to version 9.5. The progress is displayed on

the console and is logged in the file **{WCM installation directory}\installation\installation.log**.

After the configuration has been adapted, some files are copied. This may take some time.

5. If you have upgraded an Administration server, start this server and install the 9.5 license (see section “Updating the License” on page 115).

Upgrading the Data Storage

Notes

- The upgrade program changes the database tables of the WCM system. Make sure to manually back up the database (tables, procedures, packages, triggers) before starting the upgrade.
- If you use a distributed WCM system with proxy Content servers and separate data storages, not all proxy data storages may be upgraded centrally via the master Admin server. In this case, perform the data storage upgrade on the respective servers.

Prerequisites

- The data structure of the website to be upgraded corresponds to version 8.1.1 or higher.
- The upgrade of the version must have been performed successfully for at least the Admin server and the master Content server.
- The servers using the data storages to be upgraded are in the run level “Server down”.
- The assigned Admin server must be in run level “Server up”.

- For the upgrade of the WCM system, the database user requires the same rights and assignments as for the installation (see section “Privileges and assignments of the user” on page 27).
- The tablespace of the database used must have sufficient storage space. We recommend at least 20% free tablespace.

Procedure

To upgrade the data storage by means of the graphical user interface of the installation program:

1. Start the installation program (see section “Starting the Installation” on page 69), and select the installation directory of the existing WCM system.
2. In the *Type of installation* dialog box, click the *Data storage upgrade* radio button and click the *Next* button.
3. If you start the upgrade from a host computer on which no Admin server is installed, a connection to the Admin server must be established. In the *Admin server* dialog box, enter the parameters of the responsible Admin server. This is necessary for establishing a connection to this server.

   You can check these parameters in the Admin client via *Configuration* → *Pools* → *WCM* → *{Admin server pool}*.
4. Click the *Check* button.
5. You must log in to the Admin server. Enter the user ID and the password of the responsible administrator.
6. Confirm the *Admin server* dialog box by clicking the *Next* button.
7. The upgrade program tries to connect to all JDBC pools that exist in the configuration. This is independent of the assignment of pools to proxy servers.

To establish the connection to the database, the upgrade program accesses the directory **{WCM installation directory}\external_lib** and searches the JAR and/or ZIP files located in this directory for JDBC drivers.

If the connection has been established successfully, the version of the existing data structures is read. If the connection cannot be established, this is indicated for the respective pool under *Version or note*.

The following dialog box displays the data storages that can be upgraded.

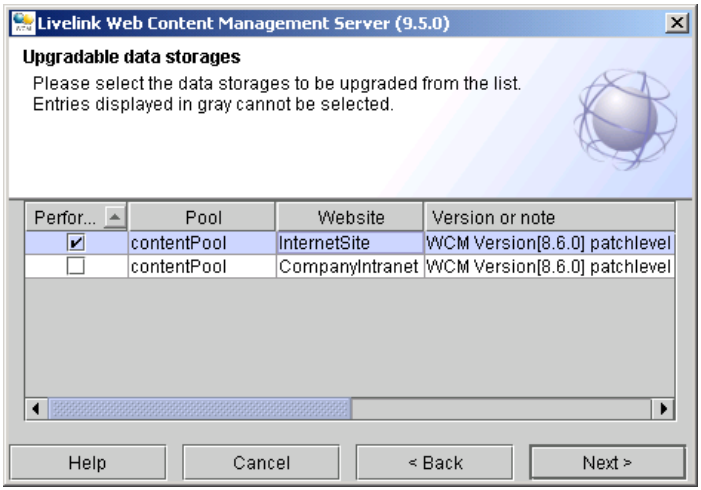


Fig. 26 – Upgrading the Data Storage

8. Select the check box for the pool/website combination to be upgraded.

   The *Selected data storages* dialog box gives you an overview of the data storages that will be upgraded.

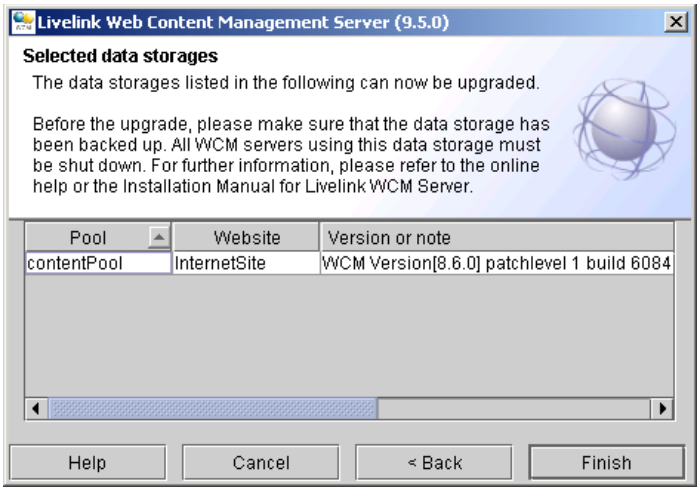

Livelink Web Content Management Server (9.5.0)
Selected data storages
The data storages listed in the following can now be upgraded.
Before the upgrade, please make sure that the data storage has been backed up. All WCM servers using this data storage must be shut down. For further information, please refer to the online help or the Installation Manual for Livelink WCM Server.
Pool
Website
Version or note
contentPool
InternetSite
WCM Version[8.6.0] patchlevel 1 build 6084
Help
Cancel
< Back
Finish

Fig. 27 – Data storages selected for upgrade

9. Click the *Finish* button.

   After the website has been adapted to the data structure of version 9.5, the respective Content server can be restarted. The website can now be accessed.

Important! If errors occur while upgrading the data storage, the changes already made cannot be undone. In this case, restore the database backup.

Upgrade via Console

As an alternative to the graphical user interface, you can also upgrade the WCM system via the console. The individual steps and subsequent work correspond to the upgrade via the graphical user interface.

Controlling the Version Upgrade via the Console

Notes:

Please also refer to the notes and prerequisites in section “Upgrading the Version” on page 214.

For information on starting the upgrade via the console, refer to section “Starting the Installation via the Console” on page 149.

To upgrade the version in the first step, you must modify the file **defaults.xml** (located in the directory **\installation** on the WCM CD). This refers to the sections described below. The entries for the other sections are automatically read from the installed system.

Entries in the `<common>` **section**

This section must be filled in completely (see section “Entries in the `<common>` Section” on page 131). Please note that the entry `<install_action>` must be set to the value `none`.

Entries in the `<rdbms>` **section**

In this section, the entry `<check>` must be set to the value `false`.

Entries in the <update_vip> section

This section must be filled in.

```
<update_vip>
      <update_ldap>false</update_ldap>
      <install_action>update</install_action>
</update_vip>
```

The following table explains the individual entries.

Table 22 – Entries in the <update_vip> section

Entry in the XML file	Explanation
<update_ldap>false</update_ldap>	If you used version 8.1 with an LDAP-based user administration in the past, the LDAP data structure may have to be modified. Possible values: true (adapt the LDAP data structure), false (do not adapt the LDAP data structure)
<install_action>update</install_action>	Upgrade option for the WCM system Possible values: update (upgrade the version), none (do not upgrade the version)

The output on the console contains information on the current and future (after the upgrade) versions of Livelink WCM Server.

```
------------------------------------------------------------------
start version update
current installed version VIP 8.6.0 QS 16 patchlevel 0 build 60302 at
2004-06-28
version to install VIP 8.7.0 QS 35 patchlevel 0 build 62422 at 2004-11-01
------------------------------------------------------------------
Created directory: D:\wcm|backup_1099398160677
Xcopy: D:\wcm|config to: D:\wcm|backup_10993981606
```

```
update started
update : current master id is 79
update config/server.xml : step 80 started
update config/server.xml : step 80 finished
update : current master id is 80
...
```

Upgrading the Data Storage

Notes:

Please also refer to the notes and prerequisites in section “Upgrading the Data Storage” on page 217.

For information on starting the upgrade via the console, refer to section “Starting the Installation via the Console” on page 149.

To upgrade the data storage in the second step, you must modify the file **defaults.xml** (located in the directory **\installation** on the WCM CD). This refers to the sections described below. The entries for the other sections are automatically read from the installed system.

Entries in the <admin> section

This section must be filled in completely (see section “Entries in the <admin> Section” on page 132). Please note that the parameter <install_action> must be set to the value none.

Entries in the <update_vip> section

In this section, the entry <install_action> must be set to the value none.

Entries in the <update_rdbms> section

This section must be filled in.

```
<update_rdbms>
      <pool name="contentpool">
         <website>InternetSite</website>
      </pool>
      <install_action>update</install_action>
</update_rdbms>
```

The following table explains the individual entries.

Table 23 – Entries in the <update_rdbms> section

Entry in the XML file	Explanation
<pool name="contentpool">	Name of the JDBC pool for the database connection. The master Content server uses this database connection for saving the WCM objects of the website specified in the entry <website>.
<website>InternetSite</website>	Name of the website that uses the JDBC pool specified in the entry <pool name> and for which the upgrade is to be performed
<install_action>update</install_action>	Upgrade option for the data storage Possible values: update (upgrade the data storage), none (do not upgrade the data storage)

Steps Required after the Upgrade

Web applications

After the upgrade, new WAR files must be generated via the Admin client for all servers integrated as web application in the application server (For more information, refer to the Livelink WCM Server Administrator Manual).

There are two ways of deploying the generated WAR files on the application server:

- by means of the functionality of the application server. Please note that some application servers delete the directory of the web application before redistributing a web application. In this case, new deployment systems must be created.
- by extracting the WAR file and copying the extracted files to the existing directory of the web application. Changes that you made to the file **web.xml** must be made again.

Server start scripts

After a successful upgrade, the server start scripts must be modified manually with regard to the following aspects:

- use of a higher version of the Java 2 SDK
- adding new copy commands

  The copy commands ensure that files located in the directory **{WCM installation directory}\latestpatch** of a server are copied to the server's **\lib** directory before the server starts. This way, Service Packs can be easily installed later.

  For a **Windows**-based system: before the line

```
call "{WCM installation directory}\setClasspath.bat"
```

enter the following lines:

```
REM copy latest patch files
java -cp .\lib\vipcore.jar de.gauss.io.FileCopy .\latestpatch\
                                                        .\lib jar
```

If the server is an Admin server, insert the following lines:

```
REM copy latest patch files
java -cp .\lib\vipcore.jar de.gauss.io.FileCopy .\latestpatch\
                                                        .\lib jar
java -cp .\lib\vipcore.jar de.gauss.io.FileCopy .\latestpatch\
                                                  .\admin\lib jar
```

For a **UNIX**-based system: after the verification whether the WCM process has been started as root, the following lines must be inserted:

```
for i in `ls ./lib/` ;do
   cp ./latestpatch/$i ./lib/$i 2>deleteme.txt
done
if [ -f deleteme.txt ]
then rm deleteme.txt
fi
```

Note: If you want to start a Content server for which no start script has been created, you can copy an existing start script and adapt the server names. Alternately, you can use the script **startserver.bat** and enter the name of the server to be started as parameter.

RDBMS Oracle

After successfully upgrading the Oracle database tables, you should update the statistics.

APPENDIX A

Product-Specific Information for LDAP Directory Services

This appendix contains information on the product-specific preparations for integrating Livelink WCM Server with the following LDAP directory services:

- Microsoft Active Directory, see the following section
- Novell eDirectory Server, see section “Novell eDirectory (NDS)” on page 236
- Sun ONE Directory Server, see section “Sun ONE Directory Server” on page 241
- OpenLDAP, see section “OpenLDAP” on page 245

For information about the general procedure for integrating an LDAP directory server with Livelink WCM Server, refer to chapter 3 “Configuring the LDAP Directory Service”.

Microsoft Active Directory

The following must be considered for the directory service Microsoft Active Directory.

Object classes

The object class `vip` must be created as abstract basic class for all WCM-specific object classes (Active Directory object class type =”Abstract”). For the object classes `vipUser`, `vipGroup`, and `vipRole`, the ADS object class type “Auxiliary” must be selected.

After configuring the object classes, add the object class `vipUser` as an auxiliary class to the predefined Active Directory object class `user`. The object classes `vipGroup` and `vipRole` must be added as auxiliary classes to the predefined Active Directory object class `group`. If you use the option *Collective groups/roles*, you must additionally add the object classes `vipGroup` and `vipRole` as auxiliary classes to the collective object classes (e.g. `organizationalUnit`).

After defining the appropriate object class types for the WCM object classes, you can use the Admin client to extend existing entries of the object classes `user` and `group` by the WCM-specific attributes (see chapter “User Administration” in the Livelink WCM Server Administrator Manual).

You can use the Admin client to create users, groups, and roles in the LDAP server. This presupposes that you specified the object classes `user` and `group`, which are predefined for users, groups, and roles in Active Directory, during the installation of the WCM system (see section “Specifying WCM-Specific LDAP Parameters” on page 85). In the settings of the LDAP pool, the entry “cn” must be selected under *Naming attribute for user*.

Attributes

Active Directory uses the attribute `cn` as the naming attribute for the LDAP entry. The value of the naming attribute must be unique.
Livelink WCM Server is not able to evaluate multi-valued “relative distinguished names” (RDN).

The following table shows the WCM attributes, their existence in Active Directory, the respective data type, and the mapping of the WCM attributes to the LDAP attributes.

Notes on creating object classes and attributes

- If some of the listed attributes already exist in the LDAP server, they can simply be assigned to the WCM classes provided they have the right semantics and syntax.
- If some of the attributes that already exist in the LDAP server have valid values, but different names, assign the required WCM attributes to the existing LDAP attributes. This is called *mapping*. The default mapping pairs are listed in the following table. For information on the mapping procedure, refer to section “Mapping WCM Attributes to LDAP Attributes” on page 42.
- If there are attributes that already exist in the LDAP server and have the same name, but invalid values, you must create additional user-defined attributes.
- The “Single value” column indicates whether the WCM system expects the attribute to be a single value. If there is a check mark in this column, the attribute value **must** be single-valued. In the LDAP directory service, “Multi-value” may still be set as type of the attribute. You must, however, ensure that the attribute has only one value. Otherwise, it cannot be guaranteed that Livelink WCM Server correctly evaluates the attribute.

- The “Mandatory” column indicates whether the WCM system expects the attribute to have a value. If there is a check mark in this column, the attribute **must** have a value.

Table 24 – WCM attributes for the class `vip` (Active Directory)

WCM attribute	Data type (syntax)	Exists in Active Directory	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`vipAccess`	case ignore string			✓	✓
`vipWebsite`	case ignore string				
`vipFuncarea`	case ignore string				
`vipRights`	case ignore string				✓
`vipType`	case ignore string			✓	✓

Table 25 – WCM attributes for the class `vip` (Active Directory)

WCM attribute	Data type (syntax)	Exists in Active Directory	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`cn`		✓		✓	✓
`uid`		✓	`uid` → `samaccountname`	✓	✓
`email`		✓	`email` → `mail` (default mapping)	✓	✓
`vipLanguage`	case ignore string			✓	✓
`vipUserpassword`		✓		✓	✓
	Note: Please note the information in section “Extended configuration for Active Directory” on page 234.				
`initPassword`	case ignore string			✓	✓
`trustedLogin`	case ignore string			✓	✓
`vipSubstitute`	DN				✓
`hclProfiles`	case exact string				
`vipDomain`	case ignore string				

Table 26 – WCM attributes for the classes `vipGroup` and `vipRole` (Active Directory)

WCM attribute	Data type (syntax)	Exists in Active Directory	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`cn`		✓		✓	✓
`member`		✓			
`email`		✓	`email` → `mail` (default mapping)		✓

Note: By means of so-called *inverse LDAP attributes*, such as `memberof`, you can speed up searches for user data, see section “Speeding up LDAP Requests” on page 51.

Extended configuration for Active Directory

Certain functions require special mapping entries for Active Directory. You can make these entries in the **defaults.xml** file (before the installation) or in the **ldapmapping.xml** file (after the installation). See section “Mapping WCM Attributes to LDAP Attributes” on page 42. The following is an overview of these mapping entries:

- changing the password via Livelink WCM Server

```
<USER_PASSWORD>
   <vipattr>vipUserpassword</vipattr>
   <ldapattr>unicodePwd</ldapattr>
   <ldapread>false</ldapread>
   <ldapwrite>true</ldapwrite>
   <codec>de.gauss.vip.jndi.codec.ADSUnicodePwd</codec>
</USER_PASSWORD>
```

> **Note:** For passwords to be changed via Livelink WCM Server, the communication to the LDAP server must be performed via a secure connection (SSL). If an SSL connection is not possible, set the tag `<ldapwrite>` to the value `false`.

- creating groups and roles via Livelink WCM Server

```
<ADS_ACCOUNTNAME>
   <vipattr>sAMAccountName</vipattr>
   <ldapattr>sAMAccountName</ldapattr>
   <ldapread>false</ldapread>
   <ldapwrite>true</ldapwrite>
   <ldapdef>$cn</ldapdef>
</ADS_ACCOUNTNAME>
```

- activating user accounts

  In Active Directory, user accounts created via LDAP are usually deactivated. For the accounts to become active immediately, the following entry is required:

```
<ADS_ACCOUNTCONTROL>
   <vipattr>userAccountControl</vipattr>
   <ldapattr>userAccountControl</ldapattr>
   <ldapread>false</ldapread>
   <ldapwrite>true</ldapwrite>
</ADS_ACCOUNTCONTROL>
```

Novell eDirectory (NDS)

Notes

- The configuration described in the following is based on the assumption that the administration of the WCM users is realized on the basis of an existing NDS directory service. If you use Novell eDirectory exclusively for managing the WCM users and not for managing the users of the company network, different settings may be necessary. We recommend that you cooperate with the Professional Services Group of Gauss Interprise AG.
- If Livelink WCM Server does not use SSL for accessing the LDAP directory service, the passwords are transmitted in plain text. This must be enabled in the configuration of Novell eDirectory. You can make this setting in the NDS Administration Console (ConsoleOne) by selecting the root context and choosing *LDAP Group* → *Properties* → *General* tab in the right window pane.
- Access to Novell eDirectory by other systems, such as Livelink WCM Server, must be activated in the configuration of NDS. Refer to the Novell eDirectory documentation for according information.

For the directory service Novell eDirectory, the following must be considered when configuring object classes and attributes.

Object classes

The object class `vip` must be created as abstract basic class for all WCM-specific object classes (NDS object class type =“Non-Effective”). For the object classes `vipUser`, `vipGroup`, and `vipRole`, the NDS object class type “Auxiliary” must be selected.

After defining the appropriate NDS object class types for the WCM object classes, you can use the Admin client to extend existing NDS entries by

the WCM-specific attributes (see chapter “User Administration” in the Livelink WCM Server Administrator Manual).

> **Note:** You **cannot** use the Admin client to create users, groups, and roles in the LDAP server. New users, groups, and roles must be created via ConsoleOne.

Installation

When installing the WCM system, the attribute `objectclass` must be selected for storing the principal type (see section “Setting the Parameters for the LDAP Directory Service” on page 82).

Attributes

NDS can use the attribute `cn` or `uid` (also `uniqueID`) as naming attribute for the LDAP entry. For uniqueness purposes, it is advisable to use the attribute `uid`.

The following table shows the WCM attributes, their existence in NDS, the respective data type, and the mapping of the WCM attributes to the LDAP attributes.

Notes on creating object classes and attributes

- If some of the listed attributes already exist in the LDAP server, they can simply be assigned to the WCM classes provided they have the right semantics and syntax.
- If some of the attributes that already exist in the LDAP server have valid values, but different names, assign the required WCM attributes to the existing LDAP attributes. This is called *mapping*. The default mapping pairs are listed in the following table. For information on the mapping procedure, refer to section “Mapping WCM Attributes to LDAP Attributes” on page 42.

- If there are attributes that already exist in the LDAP server and have the same name, but invalid values, you must create additional user-defined attributes.
- The “Single value” column indicates whether the WCM system expects the attribute to be a single value. If there is a check mark in this column, the attribute value **must** be single-valued. In the LDAP directory service, “Multi-value” may still be set as type of the attribute. You must, however, ensure that the attribute has only one value. Otherwise, it cannot be guaranteed that Livelink WCM Server correctly evaluates the attribute.
- The “Mandatory” column indicates whether the WCM system expects the attribute to have a value. If there is a check mark in this column, the attribute **must** have a value.

Table 27 – WCM attributes for the class `vip` (NDS)

WCM attribute	Data type (syntax)	Exists in NDS	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`vipAccess`	boolean			✓	✓
`vipRights`	case ignore string				✓
`vipWebsite`	case ignore string				
`vipFuncarea`	case ignore string				
`vipType`[a]	case ignore string				✓

(a) If you do not use or define the attribute `vipType`, write access for this attribute must be switched off. This is done by means of the respective mapping entries in the **defaults.xml** file (`<ldapwrite>false</ldapwrite>`), see section "Controlling readability and writability of the LDAP attributes" on page 48.

Table 28 – WCM attributes for the class `vipUser` (NDS)

WCM attribute	Data type (syntax)	Exists in NDS	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`cn`		✓		✓	✓
`uid`		✓	`uid` → `uniqueId`	✓	✓
`email`	case ignore string	✓	`email` → `mail`[a] (default mapping)	✓	✓
`vipLanguage`	case ignore string			✓	✓
`vipUserpassword`[b]		✓	`vipUserpassword` → `userPassword` (default mapping)	✓	✓
`initPassword`	boolean			✓	✓
`trustedLogin`	boolean			✓	✓
`vipSubstitute`	DN (with attribute synchroniza-tion)				✓

WCM attribute	Data type (syntax)	Exists in NDS	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
hclProfiles	case exact string				
vipDomain	case ignore string				

(a) In the GUI of ConsoleOne, the name “Internet EMail Address” is used for this attribute. Novell eDirectory internally maps this name to the LDAP name “mail”. For this reason, the mapping in the mapping file **defaults.xml** must correspond to the table, i.e. the LDAP name of the attribute must be used.

(b) For Novell eDirectory, read access for the vipUserpassword attribute must be switched off. This is done by means of the respective mapping entries in the **defaults.xml** file (<ldapread>false</ldapread> and <ldapwrite>true</ldapwrite>), see section “Controlling readability and writability of the LDAP attributes” on page 48.

Table 29 – WCM attributes for the classes vipGroup and vipRole (NDS)

WCM attribute	Data type (syntax)	Exists in NDS	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
cn		✓		✓	✓
member		✓	member → uniqueMember[a]		
email	case ignore string		email → mail (default mapping)		✓

(a) In the GUI of ConsoleOne, the name “Member” is used for this attribute. Novell eDirectory internally maps this name to the LDAP name “uniqueMember”. For this reason, the mapping in the mapping file **defaults.xml** must correspond to the table, i.e. the LDAP name of the attribute must be used.

Sun ONE Directory Server

The following must be considered for the directory service Sun ONE.

Object classes

The object class `vip` should be created as object class for all WCM-specific object classes (`vipUser`, `vipGroup`, and `vipRole`). Sun ONE object classes generally allow the extension of existing profiles by additional attributes and the creation of new profiles. Thus, you can use the Admin client to create users, groups, and roles in the LDAP server. Existing LDAP entries can be extended by the WCM-specific attributes (see chapter “User Administration” in the Livelink WCM Server Administrator Manual).

Attributes

The following table shows the WCM attributes, their existence in Sun ONE, the respective data type, and the mapping of the WCM attributes to the LDAP attributes.

Notes on creating object classes and attributes

- If some of the listed attributes already exist in the LDAP server, they can simply be assigned to the WCM classes provided they have the right semantics and syntax.

- If some of the attributes that already exist in the LDAP server have valid values, but different names, assign the required WCM attributes to the existing LDAP attributes. This is called *mapping*. The default mapping pairs are listed in the following table. For information on the mapping procedure, refer to section “Mapping WCM Attributes to LDAP Attributes” on page 42.
- If there are attributes that already exist in the LDAP server and have the same name, but invalid values, you must create additional user-defined attributes.
- The “Single value” column indicates whether the WCM system expects the attribute to be a single value. If there is a check mark in this column, the attribute value **must** be single-valued. In the LDAP directory service, “Multi-value” may still be set as type of the attribute. You must, however, ensure that the attribute has only one value. Otherwise, it cannot be guaranteed that Livelink WCM Server correctly evaluates the attribute.
- The “Mandatory” column indicates whether the WCM system expects the attribute to have a value. If there is a check mark in this column, the attribute **must** have a value.

Table 30 – WCM attributes for the class `vip` (Sun ONE)

WCM attribute	Data type (syntax)	Exists in Sun ONE	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`vipAccess`	boolean			✓	✓
`vipRights`	directory string				✓

WCM attribute	Data type (syntax)	Exists in Sun ONE	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
vipWebsite	directory string				
vipFuncarea	directory string				
vipType[a]	directory string				✓

(a) If you do not use or define the attribute vipType, write access for this attribute must be switched off. This is done by means of the respective mapping entries in the **defaults.xml** file (<ldapwrite>false</ldapwrite>), see section "Controlling readability and writability of the LDAP attributes" on page 48.

Table 31 – WCM attributes for the class vipUser (Sun ONE)

WCM attribute	Data type (syntax)	Exists in Sun ONE	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
cn		✓		✓	✓
uid		✓		✓	✓
email		✓	email → mail (default mapping)	✓	✓
vipLanguage	directory string			✓	✓

WCM attribute	Data type (syntax)	Exists in Sun ONE	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
vipUserpassword		✓	vipUserpassword → userPassword (default mapping)	✓	✓
initPassword	boolean			✓	✓
trustedLogin	boolean			✓	✓
vipSubstitute	DN				✓
hclProfiles	directory string				
vipDomain	directory string				

Table 32 – WCM attributes for the classes vipGroup and vipRole (Sun ONE)

WCM attribute	Data type (syntax)	Exists in Sun ONE	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
cn		✓		✓	✓
member		✓			
email		✓	email → mail (default mapping)		✓

OpenLDAP

The following must be considered for the directory service OpenLDAP.

Note

The following “includes” should be defined in the file **slapd.conf**.

```
include /usr/local/etc/openldap/schema/core.schema
include /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema
include /usr/local/etc/openldap/schema/misc.schema
include /usr/local/etc/openldap/schema/openldap.schema
```

Object classes

The object class `vip` should be created as object class for all WCM-specific object classes (`vipUser`, `vipGroup`, and `vipRole`). OpenLDAP object classes generally allow the extension of existing profiles by additional attributes and the creation of new profiles. Thus, you can use the Admin client to create users, groups, and roles in the LDAP server. Existing LDAP entries can be extended by the WCM-specific attributes (see chapter “User Administration” in the Livelink WCM Server Administrator Manual).

Attributes

The following table shows the WCM attributes, their existence in OpenLDAP, the respective data type, and the mapping of the WCM attributes to the LDAP attributes.

Notes on creating object classes and attributes

- If some of the listed attributes already exist in the LDAP server, they can simply be assigned to the WCM classes provided they have the right semantics and syntax.

- If some of the attributes that already exist in the LDAP server have valid values, but different names, assign the required WCM attributes to the existing LDAP attributes. This is called *mapping*. The default mapping pairs are listed in the following table. For information on the mapping procedure, refer to section “Mapping WCM Attributes to LDAP Attributes” on page 42.
- If there are attributes that already exist in the LDAP server and have the same name, but invalid values, you must create additional user-defined attributes.
- The “Single value” column indicates whether the WCM system expects the attribute to be a single value. If there is a check mark in this column, the attribute value **must** be single-valued. In the LDAP directory service, “Multi-value” may still be set as type of the attribute. You must, however, ensure that the attribute has only one value. Otherwise, it cannot be guaranteed that Livelink WCM Server correctly evaluates the attribute.
- The “Mandatory” column indicates whether the WCM system expects the attribute to have a value. If there is a check mark in this column, the attribute **must** have a value.

Table 33 – WCM attributes for the class `vip` (OpenLDAP)

WCM attribute	Data type (syntax)	Exists in OpenLDAP	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
`vipAccess`	directory string			✓	✓
`vipRights`	directory string				✓

WCM attribute	Data type (syntax)	Exists in OpenLDAP	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
vipWebsite	directory string				
vipFuncarea	directory string				
vipType[a]	directory string				✓

(a) If you do not use or define the attribute vipType, write access for this attribute must be switched off. This is done by means of the respective mapping entries in the **defaults.xml** file (<ldapwrite>false</ldapwrite>), see section “Controlling readability and writability of the LDAP attributes” on page 48.

Table 34 – WCM attributes for the class vipUser (OpenLDAP)

WCM attribute	Data type (syntax)	Exists in OpenLDAP	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
cn		✓		✓	✓
uid		✓		✓	✓
email		✓	email → mail (default mapping)	✓	✓
vipLanguage	directory string			✓	✓

WCM attribute	Data type (syntax)	Exists in OpenLDAP	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
vipUserpassword		✓	vipUserpassword → userPassword (default mapping)	✓	✓
initPassword	directory string			✓	✓
trustedLogin	directory string			✓	✓
vipSubstitute	DN				✓
hclProfiles	directory string				
vipDomain	directory string				

Table 35 – WCM attributes for the classes vipGroup and vipRole (OpenLDAP)

WCM attribute	Data type (syntax)	Exists in OpenLDAP	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
cn		✓		✓	✓

WCM attribute	Data type (syntax)	Exists in OpenLDAP	Mapping WCM attribute → LDAP attribute	Mandatory	Single value
member		✓			
email		✓	email → mail (default mapping)		✓

Glossary

Access control list – For each WCM object, users, groups, roles, and group-roles that have access to this object can be specified. The individual access rights are specified separately for each principal authorized to access the object. Also called ACL.

ACL – Access Control List

Address – see *URL*

API – Application Programming Interface. Livelink WCM Server offers various APIs to access the functionalities of the WCM servers: the WCM Java API, the remote API, the Portal Manager API, and WCM WebServices.

Applet – Java program embedded in a website. An applet is loaded by the server and executed by the client.

Application server – Also enterprise application server. An infrastructure that acts as middleware and/or development and runtime environment for web applications and wireless applications. An application server interacts with the API (Application Programming Interface). Thus, more flexibility is offered and high-level tasks can be performed remotely or via the Internet.

This allows a user at a web page to perform more sophisticated server interactions, such as querying a database or running other programs loaded on that server. Application servers often offer additional security features, load balancing, and failover mechanisms as well as scaling functions and interaction functions.

ASP – Active Server Pages. HTML files with specifically identified embedded JavaScript or Visual Basic Script programs that are run on the web server. The result is then sent to the client in normal HTML format.

Attributes – Special metadata that can be defined differently for each object type. Attributes are grouped in attribute sets.

Attribute set – Set of attributes. Attribute sets can be combined with object types. In this way, all WCM objects that are created on the basis of this object type can be equipped with the attributes of the assigned attribute set. Attribute sets are edited in the Admin client or in the Content client.

Cache – Temporary storage between slow and fast units that saves frequently used data. A cache is used to reduce the access time.

CGI – Common Gateway Interface. A web server interface used to run scripts or programs that generate user responses on HTML forms. CGI programs are usually located in a special directory on the HTTP server. Special URLs call such a CGI program, which in turn generates an HTML response to the request and sends it to the client.

Context ID – Object that is assigned to a user after successfully logging in to the WCM system. A context ID is always unique throughout the entire system. It thus precisely identifies a user. If a context ID is not used over a certain period of time, it expires.

Database – Structured data stock of related content that is managed by a database management system.

DBMS – Database management system. A DBMS permits controlled access to databases and their management.

Deployment – Deployment is the distribution of data. The deployment of Livelink WCM Server performs two main tasks: first, generating pages from the WCM objects stored in the database and distributing the generated files to the appropriate directories; second, notifying the WCM servers of changes in the WCM system.

Deployment system – The deployment systems generate pages from the WCM objects and distribute the generated files to the appropriate directories. From there, the files become visible for the users via an HTTP server. Deployment systems may be of various types and categories.

Deployment system category – Depending on the way of processing deployment jobs, deployment systems are assigned to various categories: *Standard deployment systems* automatically generate a new page every time a WCM object is changed. The generated pages are stored in the file system. *Dynamic deployment systems* generate the pages on the basis of user-defined settings and only when the page is requested via the HTTP server. The generated files are stored in a flat file structure. By means of *Search engine deployment systems*, you can prepare your website data for use with a search engine. *WebDAV deployment systems* are required for the use of WebDAV clients. *InSite Editing deployment systems* provide the basis for editing and adding content directly in the website – without the Content client.

Deployment system types – On the basis of the staging concept of Livelink WCM Server, a distinction is made between deployment systems of type “Edit”, “QA”, and “Production”. Different views of the website data are generated, depending on the type.

Edit view – In the Edit view of Livelink WCM Server, the objects of a website are created and edited. Here the most current status of the objects is visible.

Extranet – Information platform based on Internet technology for business communication with authorized external users, e.g. partners or customers.

Firewall – Hardware or software that monitors the data flow between a public and a private network and protects networks against unauthorized access. Livelink WCM Server supports architectures protected by firewalls.

Form – Possibility to integrate dialog elements to be filled in by users in web pages. The content of such a form can be sent to the server for further processing.

Frame object – Object type for a frameset

Frame topic – Frame object that is at the same time used as a topic (for content organization). See also *Topic*.

Group – Collection of users for which specific access rights can be defined. User groups are usually tied to organizational units, such as departments and projects.

Group-role – Combination of a group and role. The group-role is not an organizational unit that is defined in the Admin client, but a configuration option for the object access rights in the Content client.

HTTP – Hypertext Transfer Protocol. A communication protocol for transferring HTML pages

HTTP server – An HTTP server offers an HTTP client (browser) HTTP services over a standardized TCP/IP port.

HTTP tunneling – When HTTP tunneling is used, the data in a WCM system is sent wrapped in a HTTP data flow. The data is coded according to the VIPP protocol and additionally packaged in HTTP.

In firewall scenarios, HTTP tunneling is a common transfer method. Firewall systems interpret the data flow and, among other things, enable only certain protocols. If the VIPP protocol is not enabled, the data can be transmitted by means of HTTP tunneling.

Java – Object-oriented programming language developed by Sun Microsystems and used especially in the field of Internet technology. Security aspects and platform independence are the basic philosophies of Java.

Java 2 SDK – Java 2 Software Development Kit. The SDK contains all components that are required for creating and using programs and applets in Java, i.e. the Java compiler, the Java Runtime Environment, and several utilities.

JavaScript – Scripting language for integrating executable scripts in HTML pages. JavaScript can be used to incorporate plausibility checks and calculation functions in forms.

JDBC – Java Database Connectivity. A mechanism of communicating with existing databases. Drivers form the interface between the Java program and the database.

JRE – Java Runtime Environment. The JRE contains all components required for running Java programs, i.e. the Java Virtual Machine and the Java Class Library.

JSP – JavaServer-Pages. HTML files with specifically identified embedded Java programs that are converted into servlets by using the JSP engine and then executed on the web server. The result is then sent to the client in normal HTML format (without Java).

JVM – Java Virtual Machine. The JVM makes it possible to run platform-independent Java programs on a specific computer. It is part of the JDK or JRE.

LDAP – Lightweight Directory Access Protocol. The LDAP is based on the X.500 standard and is supported by most major software manufacturers. LDAP directory services are used to manage user information.

Master server – Only master servers have read and write access to the data of a WCM system. The master Content server manages website data, while the master Administration server manages the configuration and system data of the WCM system. See also *Server category*.

Metadata – Every WCM object has a number of object information assigned to it (e.g. expiration date, language). These are known as metadata.

Object category – Assignment of a WCM object to a specific category. Due to this assignment, the WCM object has a set of additional special attributes (metadata).

Object type – The specific kind of object, e.g. “HTML page”, “HTML template”, “Topic”. Various properties of the WCM object result from the object type. The object type is defined when the object is created. There

are only a few cases in which it may subsequently be changed. Object types can be edited in the Admin client or the Content client.

Pool – The different connections within a WCM system are managed in pools. These include, for example, connections for communication with an LDAP directory service or for communication between the WCM servers. If a connection is required, it is taken from the respective pool. After the data transfer, the connection is returned to the pool. Pools always combine connections of the same type, e.g. connections to databases (JDBC pools) or connections between WCM servers (WCM pools).

Portal – A portal is a website that serves the user as a central point of access – as a gate – to certain Internet services. A portal often offers topic-specific and personalized offers and information.

Production view – The Production view of Livelink WCM Server makes the released pages of a website available to the user. By means of a web server, these pages can be accessed in the Internet, intranet, or extranet.

Proxy server – A proxy server is used to intercept requests from a client application, e.g. a browser, to one or more other servers. If the proxy server can meet the request, it sends the requested data back to the client. Otherwise, it forwards the request to the specified server.

In the context of Livelink WCM Server, WCM servers of the category “proxy” do not have write access, but only read access to the WCM objects or the configuration. Changes to the WCM objects are only possible via the master Content server, changes to the configuration of the WCM system are made only via the master Administration server. See also *Server category*.

QA view – The QA view of Livelink WCM Server is used for quality assurance of the objects and thus of the website content. This view thus performs the control function between editing in the Edit view and publication in the Production view.

RDBMS – Relational database management system. A DBMS in which relations between data records from individual databases can be used. In

contrast to an RDBMS, there are also object-oriented and object-relational DBMS.

Relator – Due to the integration of web content management and Livelink, you can add Livelink objects as WCM objects in a WCM-managed website. These WCM objects are called relators. They always refer to a certain version of a Livelink object. For integrating a single Livelink object, the WCM system provides the object type “Livelink relator”. For integrating Livelink folders, the WCM object type “Livelink folder relator” is provided.

Role – Collection of users, similar to a user group, for which specific access rights can be defined. The user role is usually defined in terms of tasks, whereas user groups are generally tied to organizational units, such as departments or projects.

Search server – The combination of Index and Query system in Livelink WCM Server is called “Search server”. Each Search server is assigned to exactly one WCM server. It is, however, possible to assign more than one Search server to a WCM server.

Server category – In a WCM system, a distinction is made between master and proxy servers. Master servers have write access to the data of the WCM system, while proxy servers have only read access. The master Content server manages the website data, the master Administration server manages the configuration and system data. In addition to this, any number of proxy servers can be set up.

Server type – According to the tasks of the servers, there are two server types: Content servers for managing website data and Administration servers for managing the user, configuration, and system data of the WCM system. Basically, every Content server is able to provide all views of the data of the managed websites – Edit, QA, and Production. The available views may be limited by the fact that the Content server only receives the data of certain views.

Servlet – Java program executed by the web server for generating the data requested by a client by means of an HTTP request

SMTP – Simple Mail Transfer Protocol. A protocol for transferring e-mail messages, for example between different servers

SSL – Secure Socket Layer. A protocol layer for the communication between the components of a WCM system that ensures that the data transfer will be secure in terms of eavesdropping and falsification. SSL can be used both for the VIPP protocol and HTTP tunneling.

Statification – During statification, the dynamic components of, for example, a JSP page are converted into static components. The result is pure HTML without Java code.

TCP/IP – Transmission Control Protocol/Internet Protocol. Basic protocol for data transfer in the Internet

Topic – Combination of an HTML page for direct display of information and a list for accommodating subordinate objects. The topics are shown as nodes in the object tree and thus help structure the website.

Type – see *Object type*

URL – Uniform Resource Locator. A unique address in the World Wide Web.

VIPP – VIP Protocol. A proprietary protocol for exchanging data between the components of a WCM system. VIPP can be tunneled in HTTP for communication in WANs or over the Internet.

WCM server – In a WCM system, there are several WCM servers working in parallel (server processes). The exact tasks of a server depend on the server type and server category.

WCM tag – Special element for integrating WCM specific data in an HTML page. WCM tags are used in particular when creating templates.

WebDAV – The WebDAV (Web-based Distributed Authoring and Versioning) protocol supports Internet and group-based working on the basis of standard Internet technologies. Thanks to WebDAV, users do not need a special client for creating WCM objects, jointly editing them, and managing them by means of WevDAV-capable tools.

Web repository – The name given to the area where the WCM system stores and manages all the objects belonging to a website

Web server – Program for processing the requests from a browser. A web server corresponds to an HTTP server that offers additional web services, such as those of a JSP engine.

Web technology – Client/server technology based on TCP/IP. The advantage lies in the open standard, which permits the creation of heterogeneous systems.

Index

A

Active Directory
  configuration 230
Admin client
  installation 113
Admin server
  installation 74
administrator
  create for Livelink WCM Server in LDAP 54
  enter during installation 88
alias for Content client 177
Apache 2
  configuration 168
application server
  default application 97
  mapping 97
archive log mode (Oracle) 23
attributes (LDAP) for Livelink WCM Server 38

B

base authentication 179
  configuration in IIS 179
BEA WebLogic 8.1
  configuration 170
binding profile for LDAP directory service 54
block size (Oracle) 23
boot servers 152

C

character-encoding (Secure Access parameter) 197
collective groups/roles for LDAP 36
command line mode
  installation 126
configuration
  Apache 2 with Tomcat 168
  BEA WebLogic 8.1 170
  for Secure Access 189
  LDAP 33
  Microsoft Active Directory 230
  MS Internet Information Server with Resin 172
  MS SQL Server 29
  Novell eDirectory 236
  OpenLDAP 245
  Oracle 22
  RDBMS 21
  Sun ONE Directory Server 241
  web server 165
console
  installation 126
Content client
  alias 177
  HTTPS connection 169, 172, 178
  integrate in web application 96
  precompile script 188
  set encoding 93
Content Miner
  deinstall 121
  install 98
  set ports 100
Content server in application server
  start 159
cursors (Oracle) 24

D

data source 79
database
  create in MS SQL server 29

in general 21
parameters during installation 76
specify type during installation 78
database character set (Oracle) 23
database instances (Oracle) 23
database user
create in MS SQL server 29
create in Oracle 26
default application for application server 97
default extension for pages 76
defaults.xml for installation/ deinstallation 126
deinstallation 121
delete database tables 125
via console 126
via graphical user interface 123
directories
after installation 117
write rights 117
directory
for installation 70
directory-servlet (Secure Access parameter) 207

E

encoding
for Content client 93
for Secure Access 197
set in Oracle 23
explicit assignment (LDAP) 36

F

fallback LDAP server 56
filter mapping
for web application 97
filter-name (Secure Access parameter) 197
firewall scenario
installation 106

G

Gauss
LDAP OIDs 41

H

HTTP port 75, 92
HTTP server
configuration for Secure Access 189, 192
httpport (Secure Access parameter) 199
HTTPS
for connection to Content client 169, 172, 178
for connection to LDAP server 61

I

implicit assignment (LDAP) 36
Index system
start separately (UNIX) 161
installation
add server 110
Admin client 113
Admin server 74
configure servers 91
Content Miner 98
control 126
create WCM administrator 88
directory 70
directory structure after 117
LDAP object classes for Livelink WCM Server 85
Livelink Search server 103
Livelink WCM Server (console) 126
Livelink WCM Server (graphical user interface) 67
log 162
Lucene 101

master system behind firewall 107
minimum 68
options for license file 72
parameters of the LDAP server 82
parameters of the Livelink system 80
procedure 11
proxy Content server behind firewall 108
proxy system outside firewall 108
RDBMS parameters 76
requirements 13
start via console 149
start via graphical user interface 69
update license 115
user-defined 106
via console 126
installation log 162
inverse LDAP attributes 51

J

Java 2 SDK 14
JDBC driver
for database 78
update after patch 78
JDK 14
JSP engine
configuration for Secure Access 189, 193

L

LDAP
collective groups/roles 36
configuration 33
create binding profile 54
create WCM administrator 54
different attributes for names 49
explicit user assignment 36
fallback LDAP server 56
faster requests 51
implicit user assignment 36
map attributes for faster requests 51
map WCM attributes to LDAP attributes 42
Microsoft Active Directory 230
Novell eDirectory 236
object classes and attributes for Livelink WCM Server 38
OIDs of Gauss 41
one LDAP server for multiple WCM systems 58
OpenLDAP 245
parameters during installation 82, 85
readability and writability of attributes 48
SSL connection to LDAP server 61
standard groups/roles 36
Sun ONE Directory Server 241
use several LDAP servers 55, 57
ldapread (tag for LDAP mapping) 48
ldapwrite (tag for LDAP mapping) 48
licenses
installation options 72
update 115
Livelink
parameters during installation 80
Livelink Search server
deinstall 121
install 103
Livelink WCM Server
directory structure 118
log_checkpoint_interval (Oracle) 23
logdir (Secure Access parameter) 202
loglevel (Secure Access parameter) 201
logname (Secure Access parameter) 202
logs
installation 162
Secure Access 200

Lucene
deinstall 121
install 101

M

mail server
for Admin server 75
map WCM attributes to LDAP attributes 42
mapping for web applications 97
master system
installation behind firewall 107
memberof (LDAP attribute) 51
memory
for WCM server 154
minimum installation 68
MS Active Directory
configuration 230
MS Internet Information Server
authentication methods for Secure Access 179
configuration 173
configure base authentication 179
configure NTLM authentication 186
integrate Resin 177
MS SQL Server 2000
configuration 29
create database 29
create database user 29

N

national character set (Oracle) 23
NDS
configuration 236
new
server 110
Novell eDirectory
configuration 236
NTLM authentication 179
configuration in IIS 186

O

object classes for LDAP
create 38
open cursors (Oracle) 24
open_cursors (Oracle) 23
OpenLDAP
configuration 245
Oracle
configuration 22
configure database instances 23
create database user 26
create tablespace 26
JDBC driver 78
new JDBC driver after patch 78
open cursors 24
set UTF-8 23
owner 80

P

pages
set default extension 76
parallel_max_servers 23
parameters
defaults.xml for installation/ deinstallation 126
in server start scripts 153
patch for JDBC driver 78
pathfrom (Secure Access parameter) 198
pathto (Secure Access parameter) 198
pattern (Secure Access parameter) 206
performance
speed up LDAP requests 51
ports
for Content Miner during installation 100
for WCM server during installation

75, 92
precompile script for Content client 188
precompile the Content client 188
processes 23
profile (Secure Access parameter) 200
proxy Content server
installation behind firewall 108
proxy system
installation outside firewall 108

Q

Query system
start separately (UNIX) 161

R

RDBMS
configure for Livelink WCM Server 21
delete tables after deinstallation 125
new JDBC driver after patch 78
parameters during installation 76
Readme 118
realm(Secure Access parameter) 203
redirect-dir (Secure Access parameter) 204
redirect-url (Secure Access parameter) 204
requirements for installing Livelink WCM Server 13
Resin
configuration 174
integrate in MS Internet Information Server 177
start 178

S

Search servers
assign server 100
deinstall 121
directory structure 120
install 98
start 160
secure (Secure Access parameter) 199
Secure Access
additional parameters for the servlet 206
configuring HTTP server and JSP engine 189
configuring servlet or filter 193
configuring the JSP engine 193
integrate in HTTP server 192
integrate in MS Internet Information Server 179
log options 200
name of web server 200
parameters for servlet or filter 193
set to Unicode (UTF-8) 197
Secure Access parameter
character-encoding 197
directory-servlet 207
filter-name 197
httpport 199
logdir 202
loglevel 201
logname 202
pathfrom 198
pathto 198
pattern 206
profile 200
realm 203
redirect-dir 204
redirect-url 204
secure 199
servlet 206
servlet-name 197
setcontentlength 208
translator 210
url-pattern 205
usepath 199
viphost 199

vippport 199
welcome-file-list 208
secure connection
to Content client 169, 172, 178
to LDAP server 61
server
set up service 76, 93
server category 94
servers
add 110
assign Search server 100
configure during installation 91
deinstall 121
memory 154
remove service 124
specify category during installation 94
start 152
start scripts 153
stop 157
service
for Admin server 76
for WCM servers 93
remove 124
service for Windows
for Admin server 76
for WCM servers 93
remove 124
servlet (Secure Access parameter) 206
servlet mapping
for Secure Access servlet 206
for web application 97
servlet-name (Secure Access parameter) 197
setcontentlength (Secure Access parameter) 208
shared pool (Oracle) 23
shut down servers 157
SMTP server
for Admin server 75
speed up LDAP requests 51
SQL Server 2000
configuration 29
JDBC driver 78
SSL
for connection to Content client 169, 172, 178
for connection to LDAP server 61
for WCM server 75, 93
standard groups/roles for LDAP 36
start
Content server in application server 159
Index and Query system separately (UNIX) 161
installation via console 149
installation via graphical user interface 69
Search server 160
server 152
start scripts of servers
parameters 153
stop
WCM servers 157
substituteof (LDAP attribute) 51
Sun ONE Directory Server
configuration 241

T

tablespace
create in Oracle 26
tag libraries
directory 121
taglib mapping
for web application 97
Tomcat
configuration 168
translator (Secure Access parameter) 210

truststore of Livelink WCM Server 61

U

Unicode
for Content client 93
for Secure Access 197
set in Oracle 23
UNIX
write rights for directories 117
upgrade
data storage (console) 223
data storage (graphical user interface) 217
Livelink WCM Server 213
steps after upgrade 225
via console 221
via graphical user interface 213
WCM version (console) 221
WCM version (graphical user interface) 214
url-pattern (Secure Access parameter) 205
usepath (Secure Access parameter) 199
user management
specify type of storage 76
UTF-8
for Content client 93
for Secure Access 197
set in Oracle 23

V

viphost (Secure Access parameter) 199
VIPP port 75, 92
vippport (Secure Access parameter) 199
virtual memory 154

W

WCM servers
add 110
add as service 76, 93
assign Search server 100
configuring during installation 91
deinstall 121
memory 154
remove service 124
specify category during installation 94
start 152
start scripts 153
stop 157
WCM system
deinstall 121
WCM truststore 61
web application
generate 94
web server
configuration 165
enter name for Secure Access 200
welcome-file-list (Secure Access parameter) 208
Windows service
for Admin server 76
for WCM servers 93
remove 124
write rights for directories 117